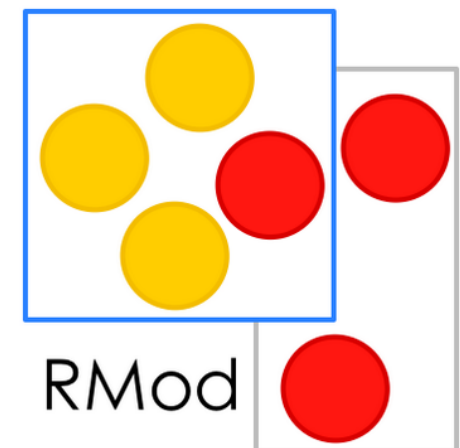


# Learn the vm

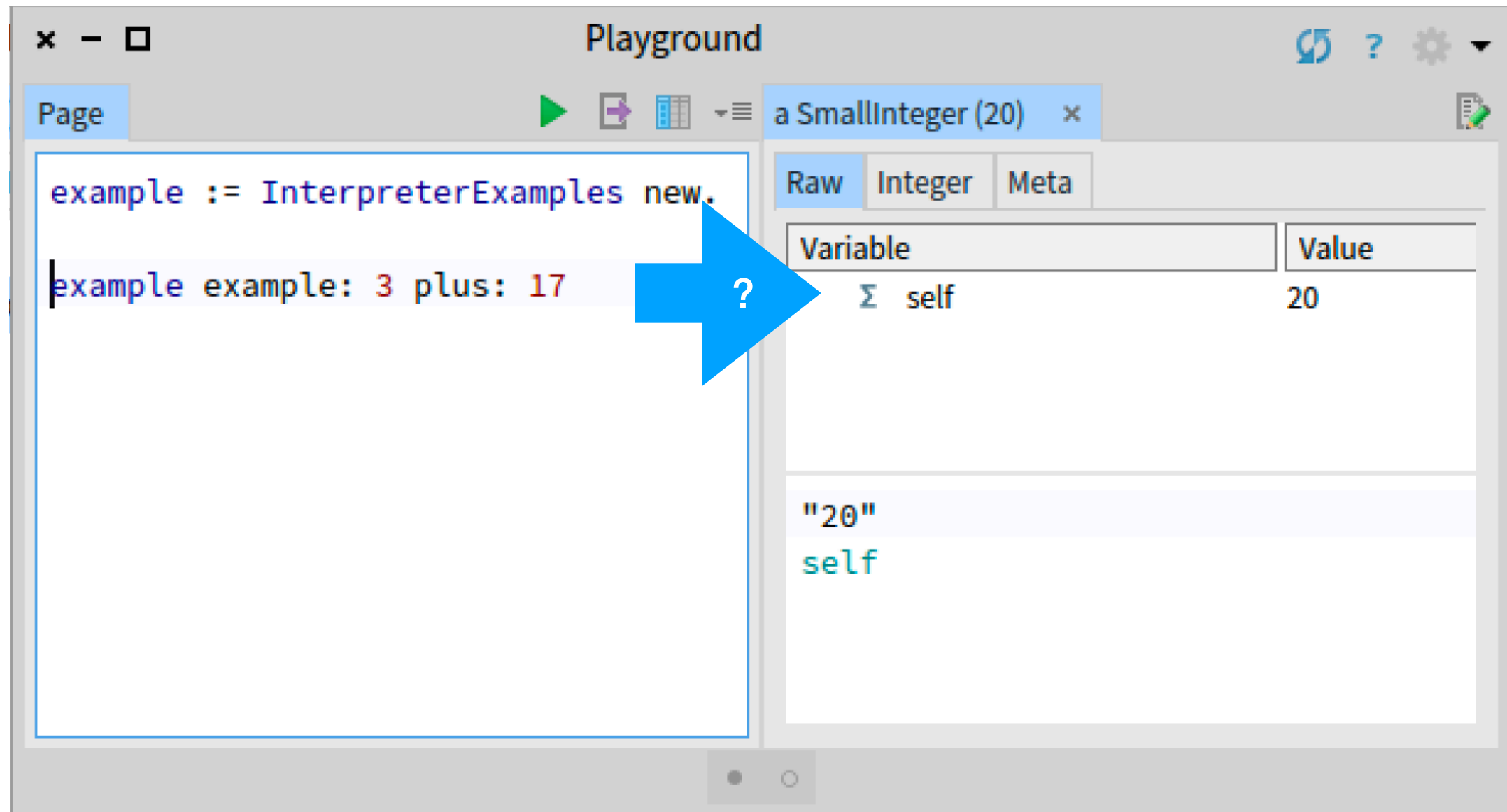
The bytecode interpreter

By Rogliano Théo

Inria, Univ. Lille, CNRS, Centrale Lille,  
UMR 9189 - CRISTAL  
Lille France



# How we get the result ?



The screenshot shows a Scala Playground window with the following code in the editor:

```
example := InterpreterExamples new.  
example example: 3 plus: 17
```

A blue arrow points from the second line of code to a variable inspection window titled "a SmallInteger (20)". The window has tabs for "Raw", "Integer", and "Meta". It contains a table with the following data:

Variable	Value
$\Sigma$ self	20

Below the table, the string representation of the value is shown as "20" and the object type as self.

Inspector on a CompiledMethod (InterpreterExamples>>#e)

a CompiledMethod (InterpreterExamples>>#example:plus:)

Raw Source Bytecode Ir AST Header Meta

```
example: aNumber plus: aNumberToAdd

^aNumber plus: aNumberToAdd
```



**Compilation**

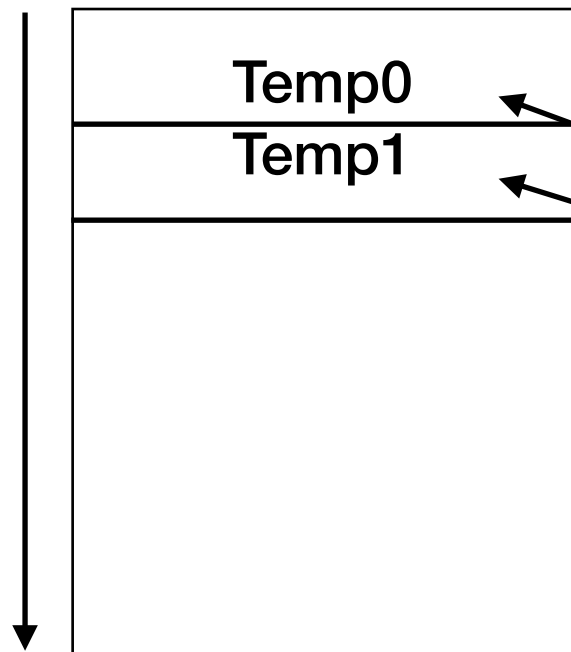
Inspector on a CompiledMethod (InterpreterExamples>>#e)

a CompiledMethod (InterpreterExamples>>#example:plus:)

Raw Source Bytecode Ir AST Header Meta

```
33 <10> pushTemp: 0
34 <11> pushTemp: 1
35 <E0> send: plus:
36 <7C> returnTop
```

# Operand Stack



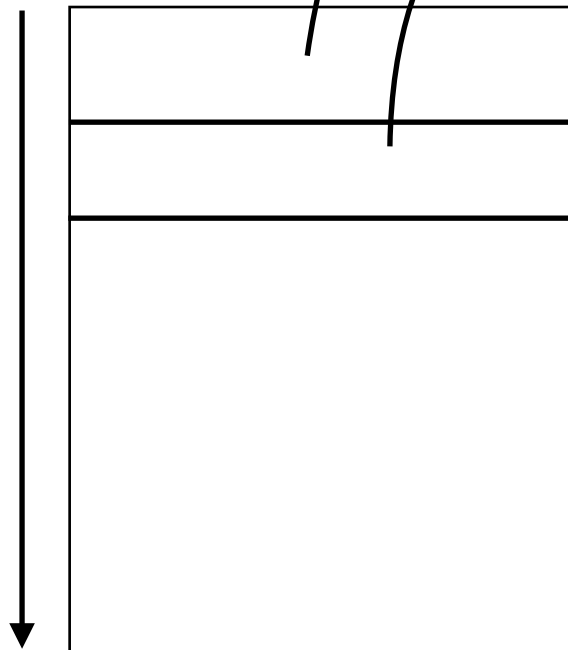
Inspector on a CompiledMethod (InterpreterExamples>>#e

a CompiledMethod (InterpreterExamples>>#example:plus:)

Raw Source Bytecode Ir AST Header Meta

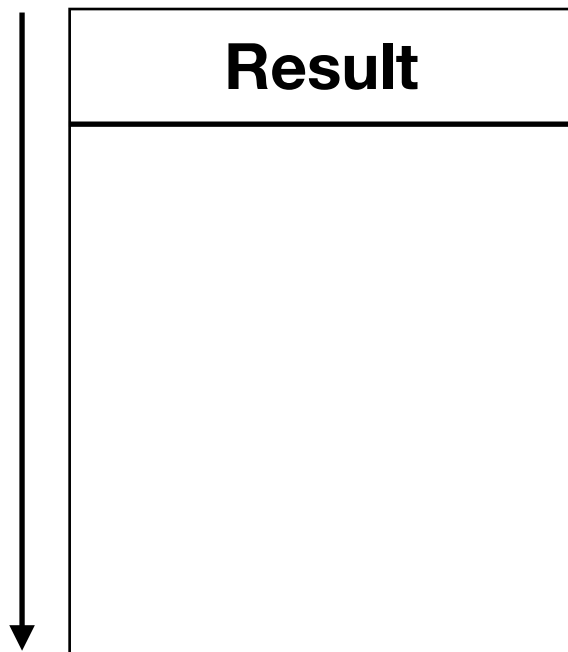
```
33 <10> pushTemp: 0
34 <11> pushTemp: 1
35 <E0> send: plus:
36 <7C> returnTop
```

# Pop



**Temp0** As receiver  
**plus:**  
**Temp1** As argument

```
Inspector on a CompiledMethod (InterpreterExamples>>#e
a CompiledMethod (InterpreterExamples>>#example:plus:)
Raw Source Bytecode Ir AST Header Meta
33 <10> pushTemp: 0
34 <11> pushTemp: 1
35 <E0> send: plus:
36 <7C> returnTop
```



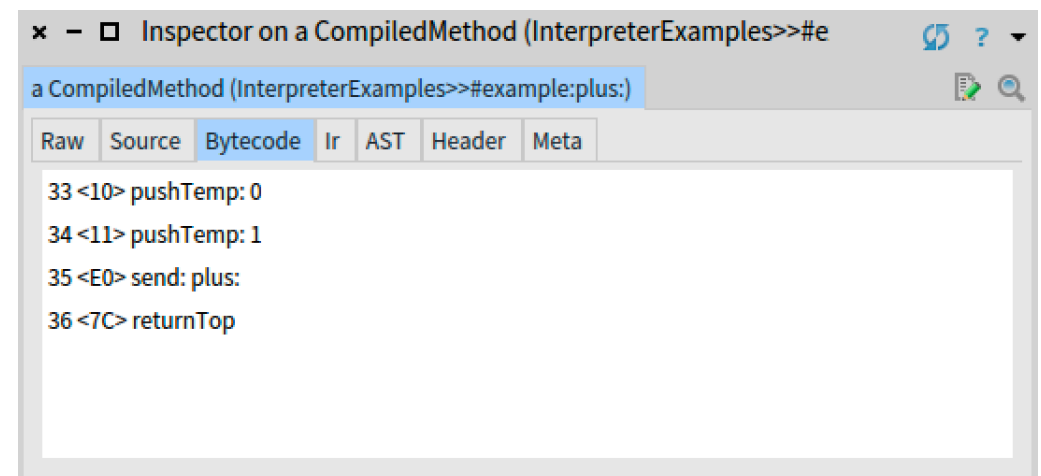
Inspector on a CompiledMethod (InterpreterExamples>>#e  
a CompiledMethod (InterpreterExamples>>#example:plus:)

Raw Source **Bytecode** Ir AST Header Meta

```
33 <10> pushTemp: 0  
34 <11> pushTemp: 1  
35 <E0> send: plus:  
36 <7C> returnTop
```

**Virtual machine**

# Memory

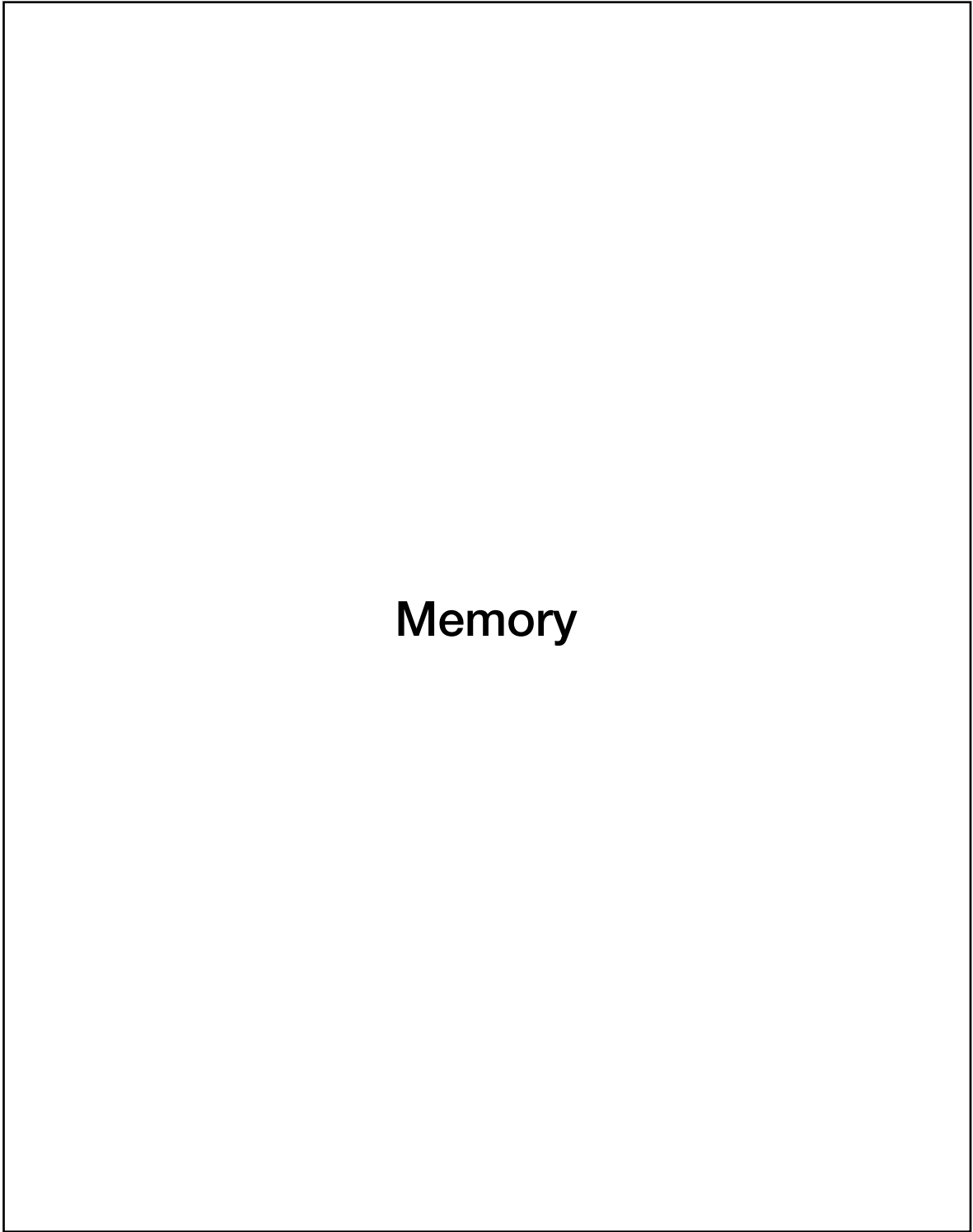
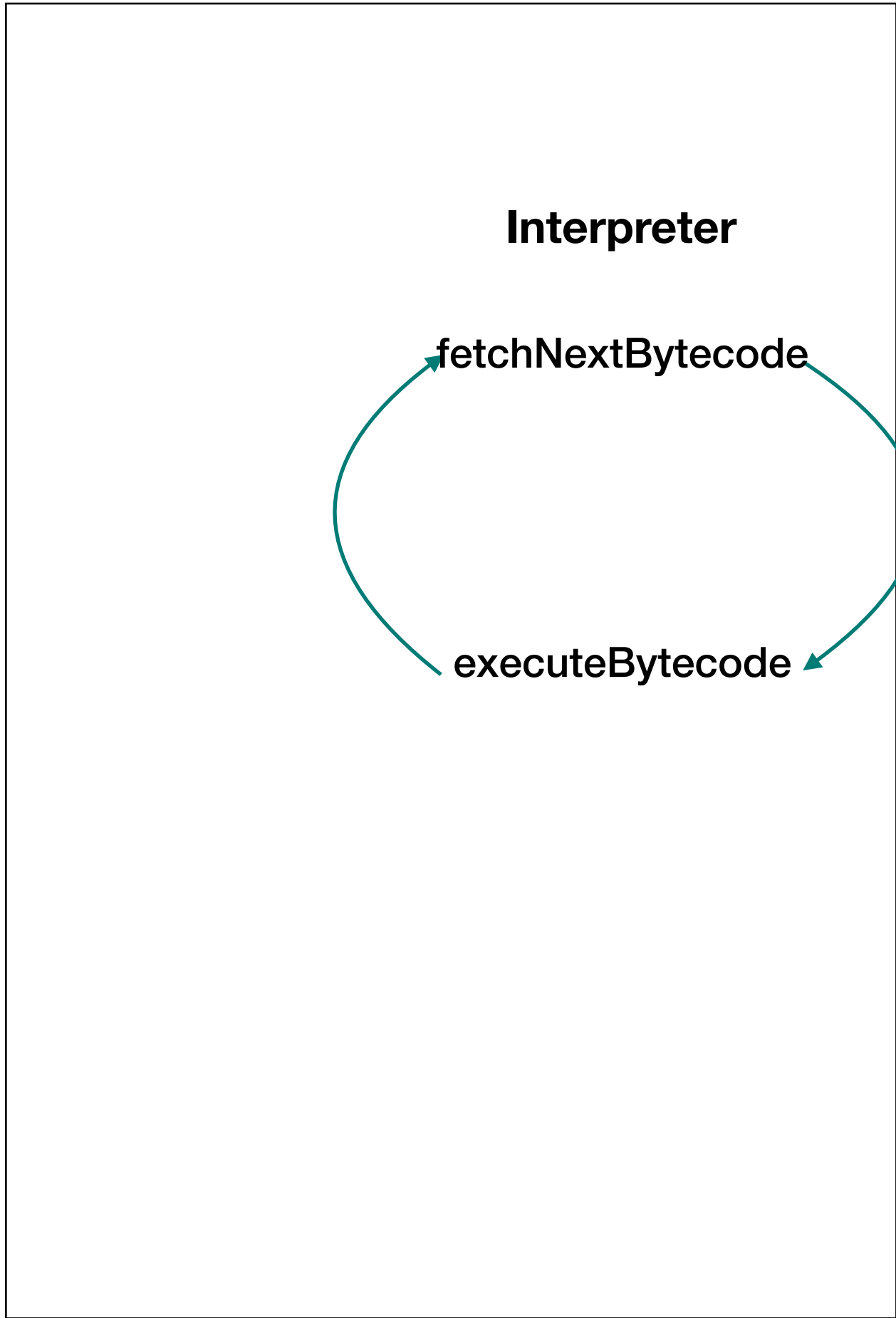


The screenshot shows a debugger window titled "Inspector on a CompiledMethod (InterpreterExamples>>#e)". The selected object is "a CompiledMethod (InterpreterExamples>>#example:plus:)", and the "Bytecode" tab is active. The bytecode instructions are as follows:

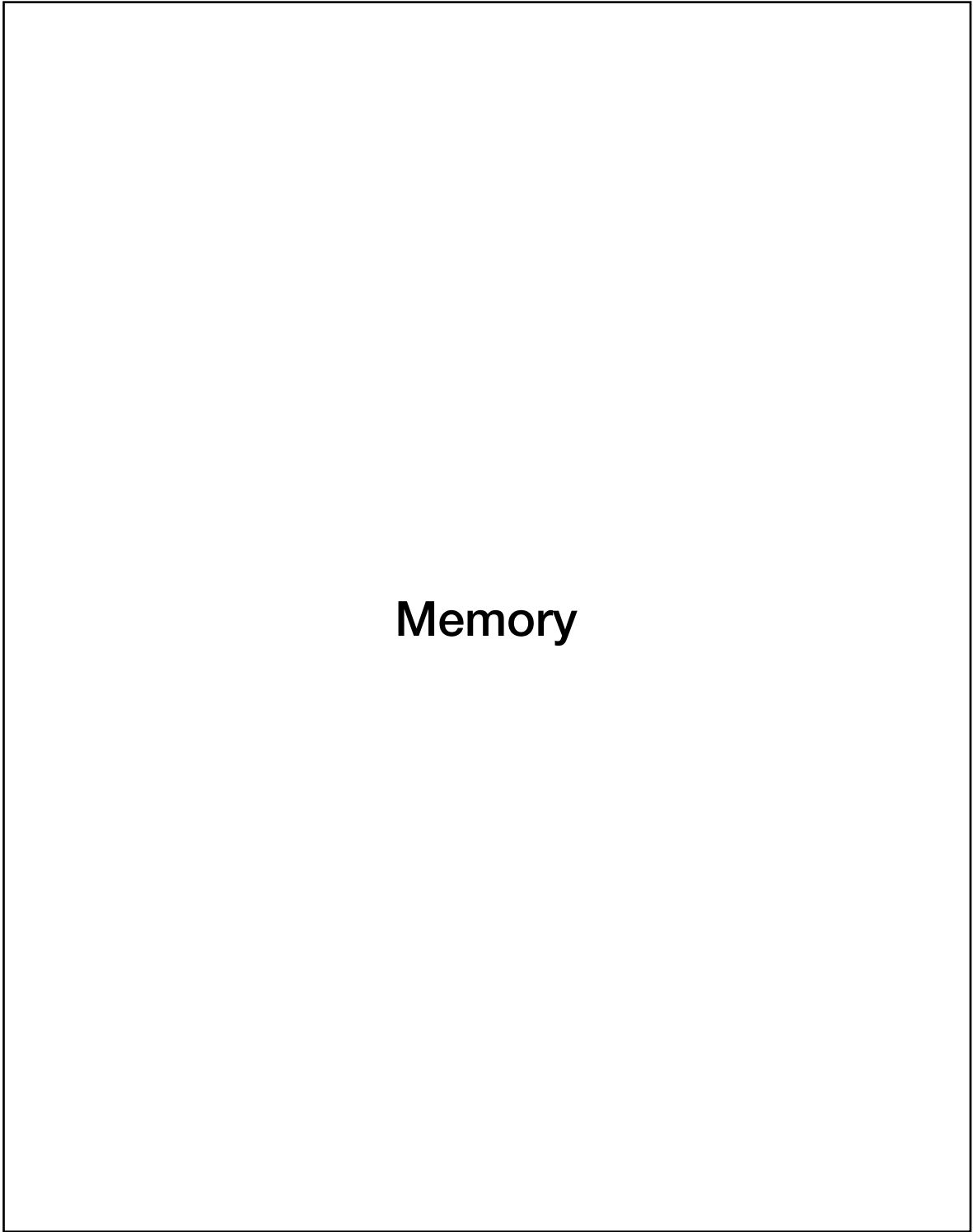
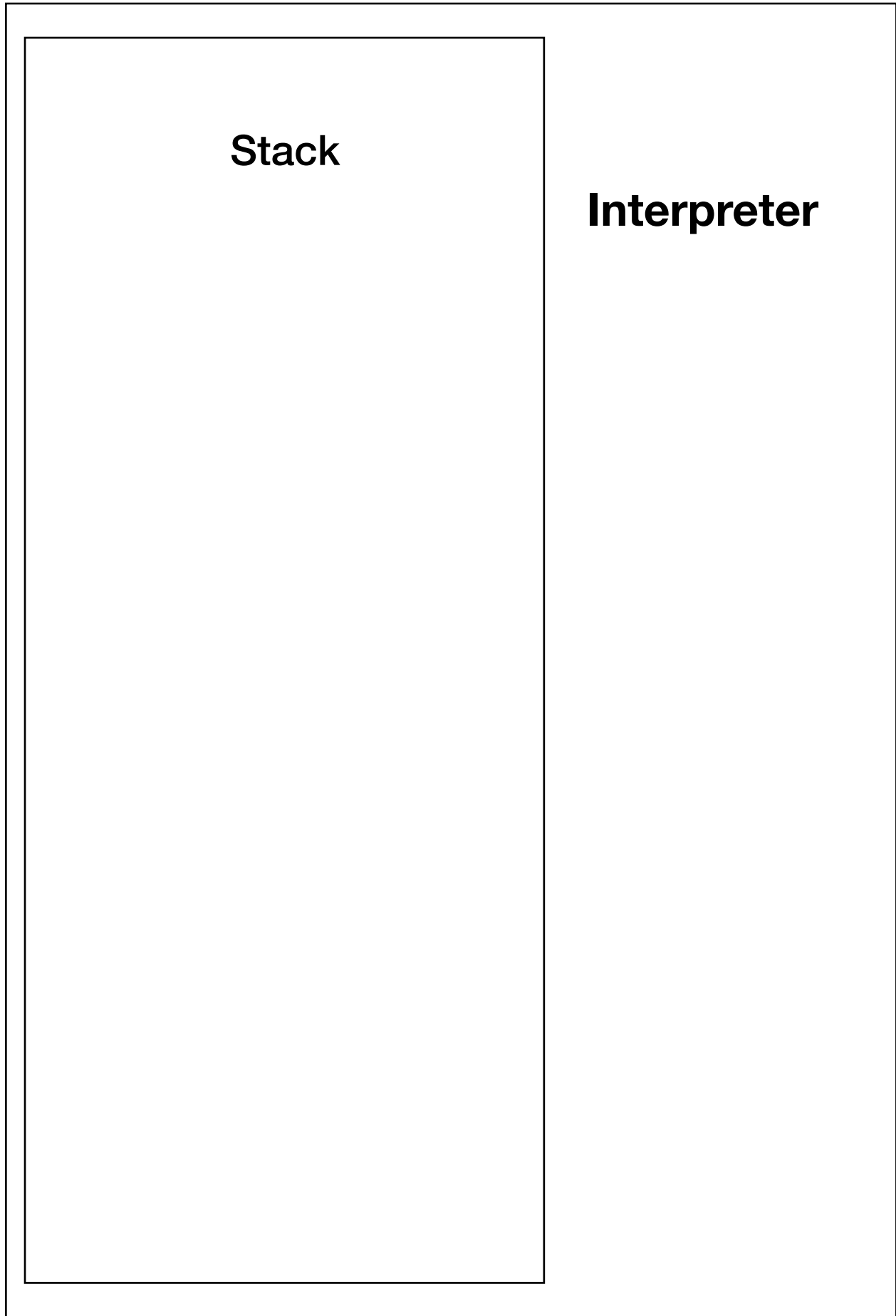
```
33 <10> pushTemp: 0
34 <11> pushTemp: 1
35 <E0> send: plus:
36 <7C> returnTop
```

# Virtual Machine

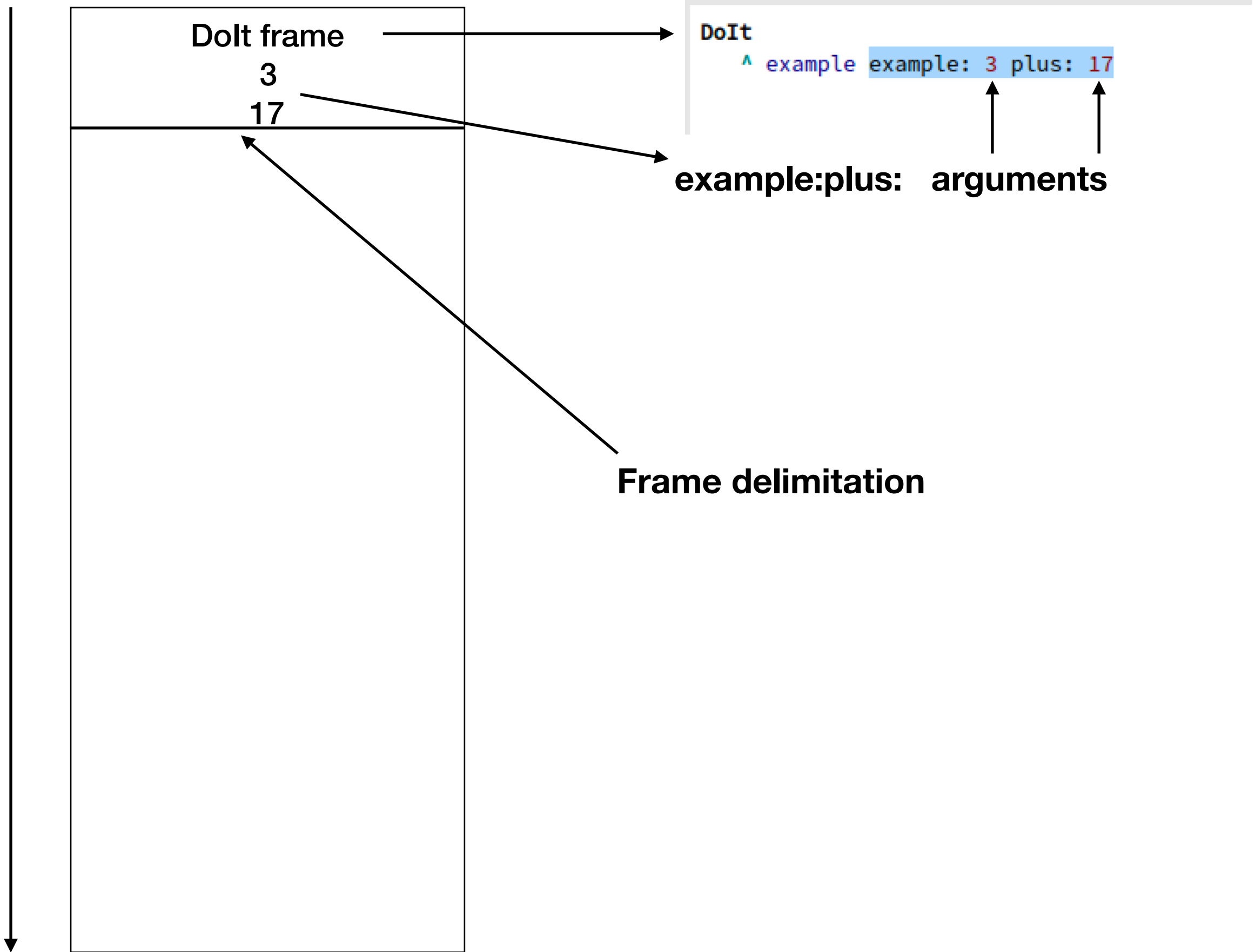


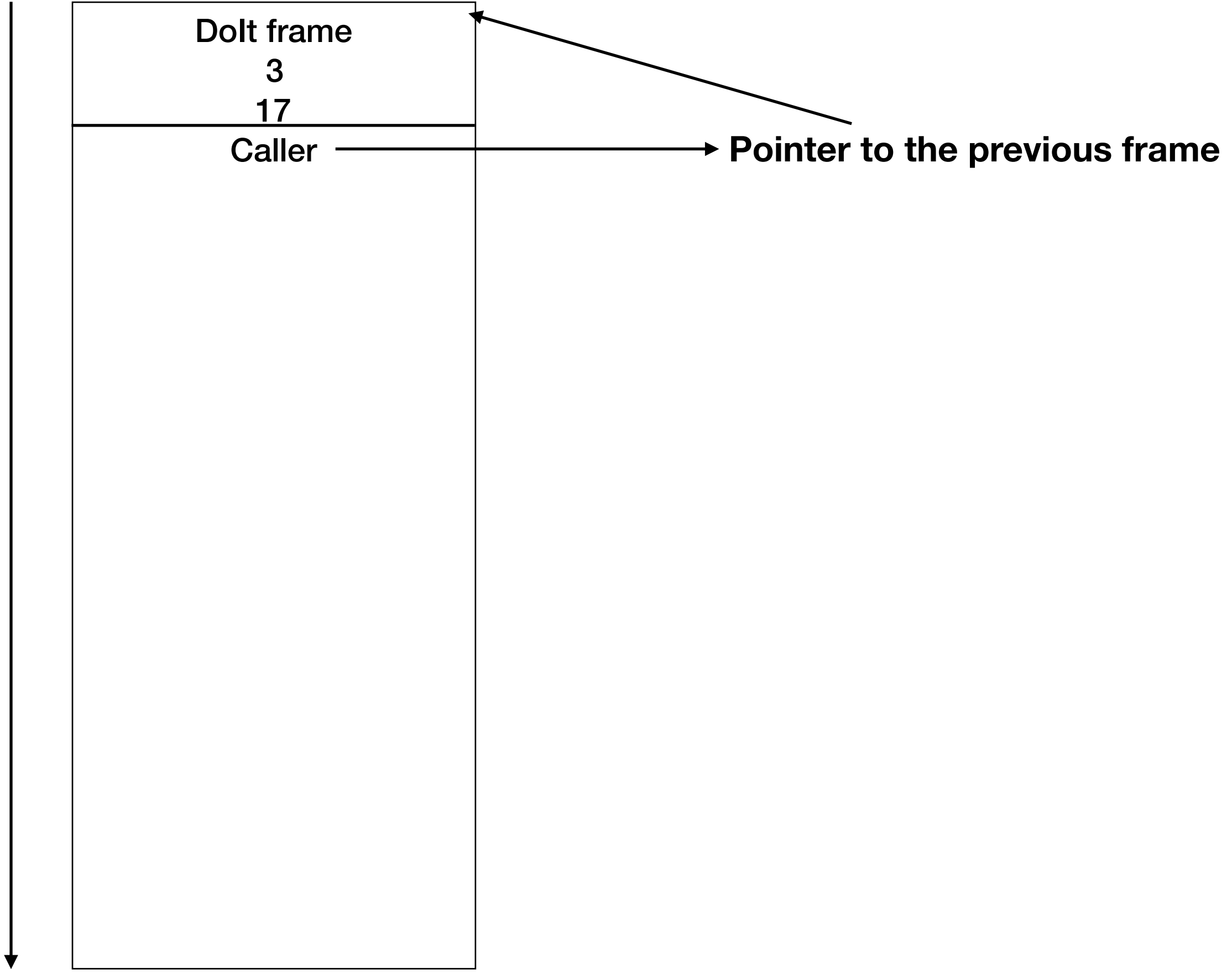


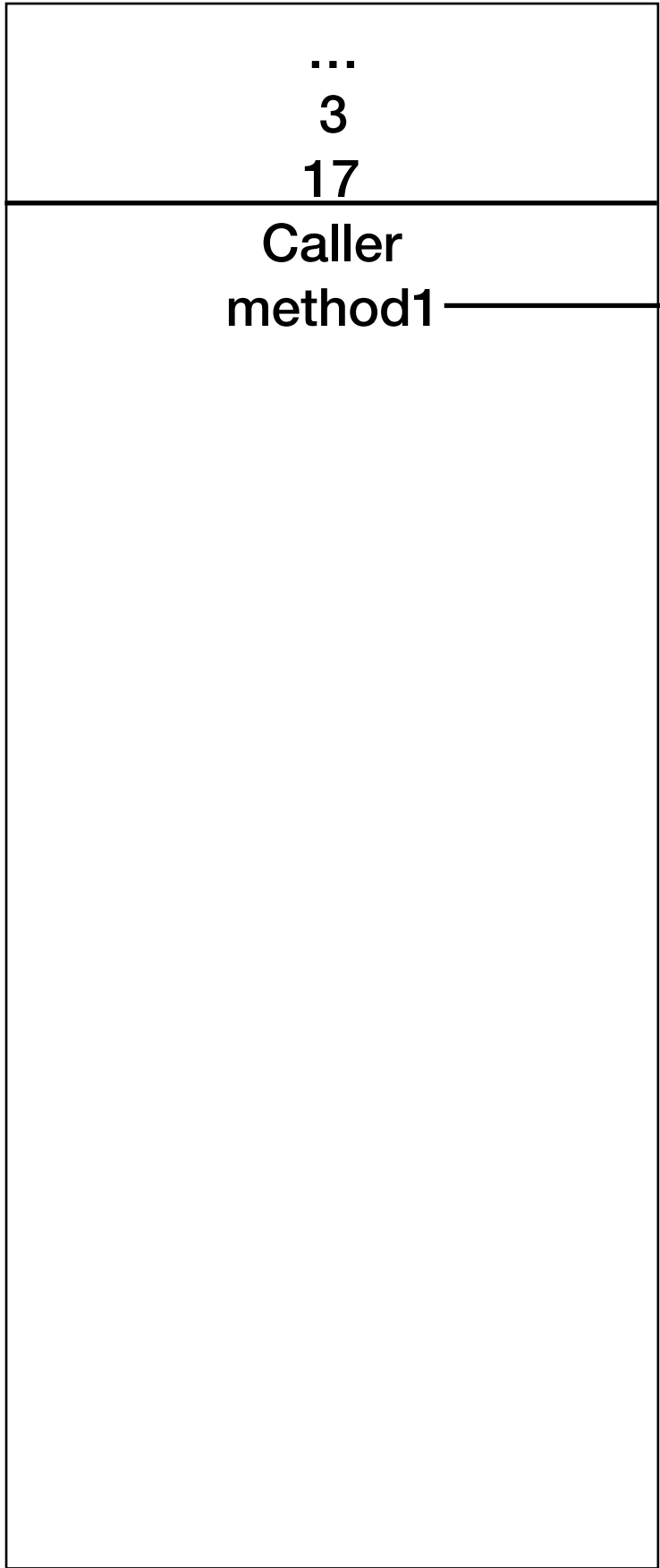
**Virtual Machine**



**Virtual Machine**



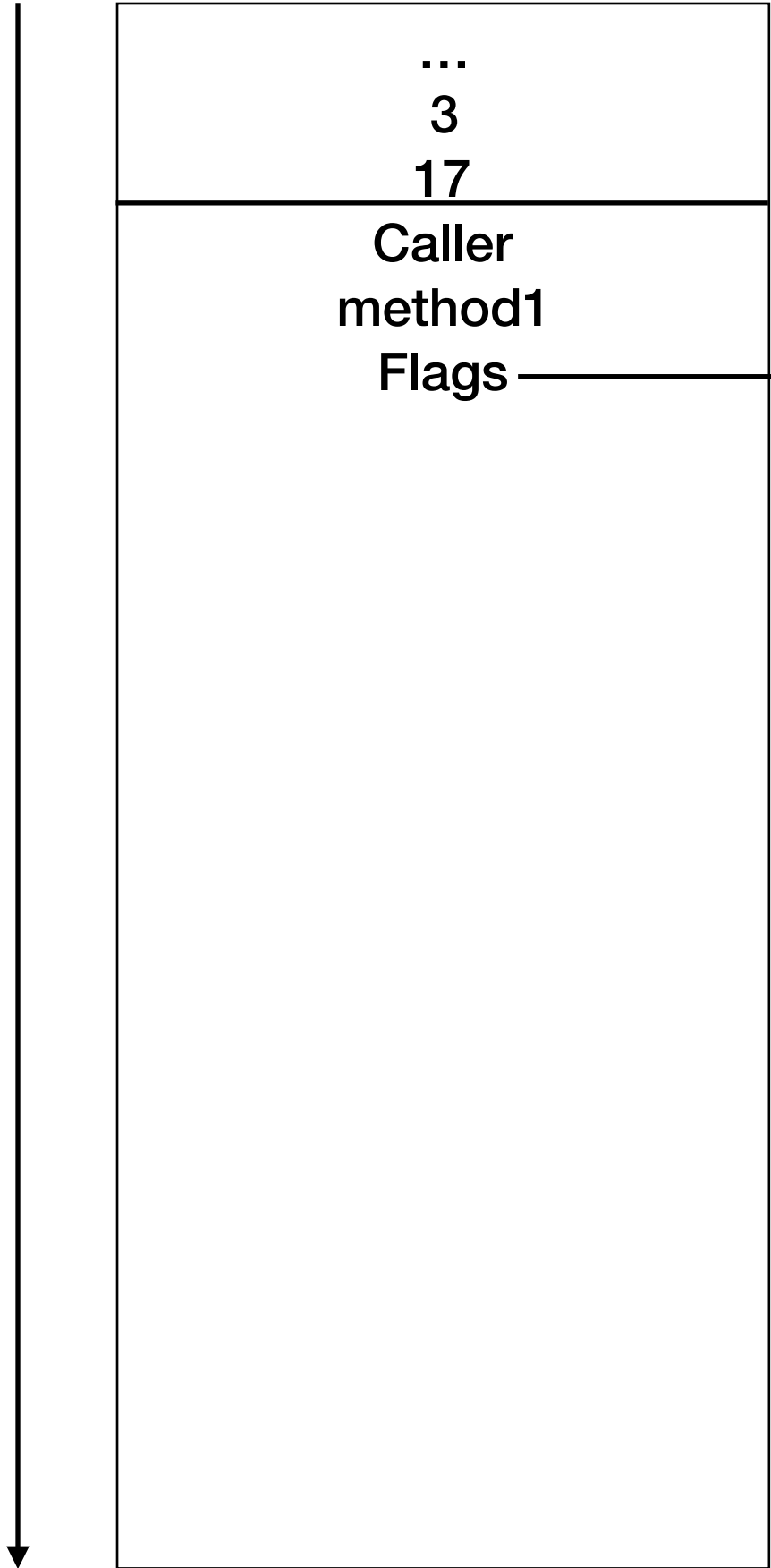




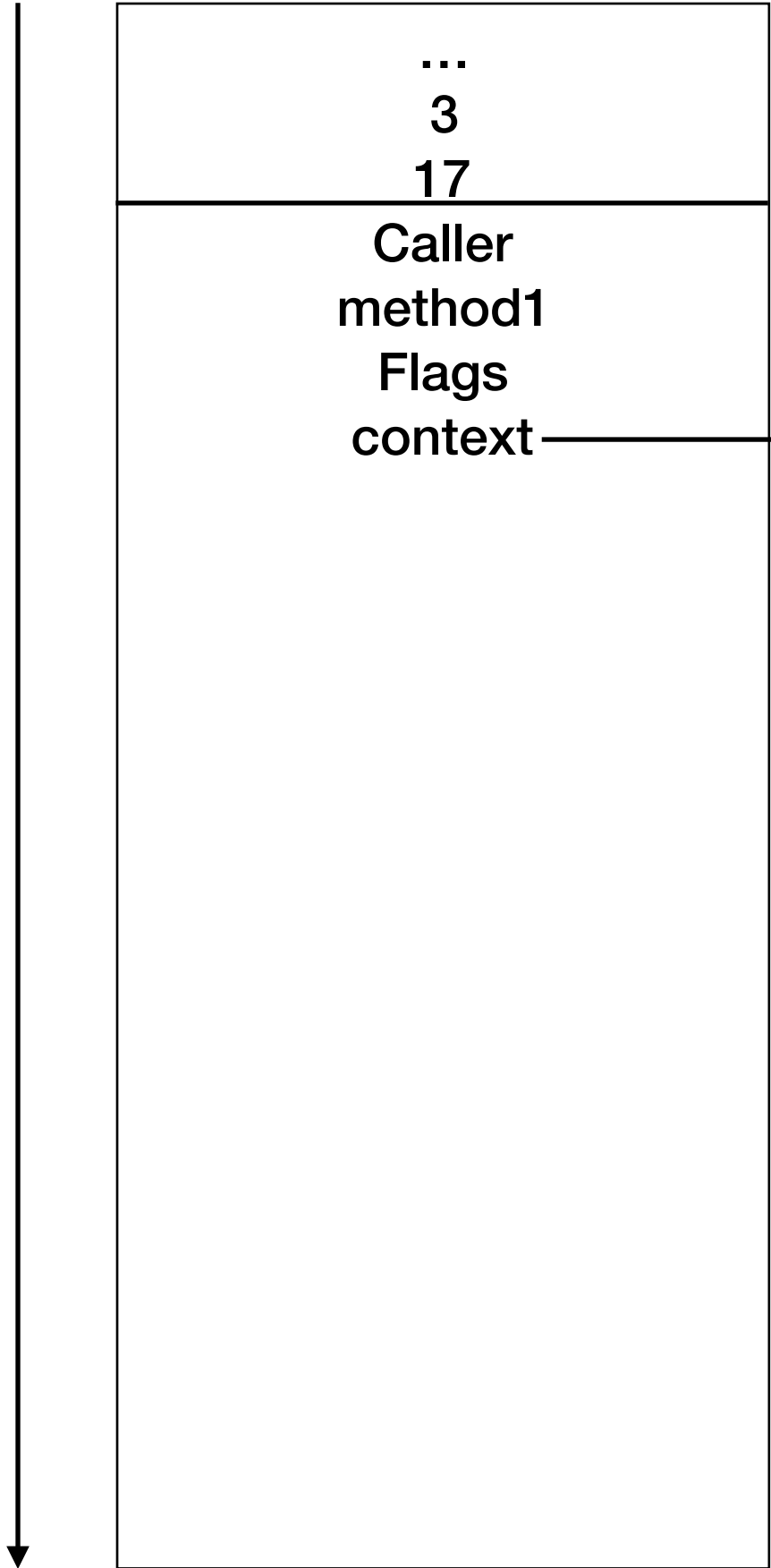
Inspector on a CompiledMethod (InterpreterExamples>>#e  
a CompiledMethod (InterpreterExamples>>#example:plus:)

Raw Source **Bytecode** Ir AST Header Meta

```
33 <10> pushTemp: 0  
34 <11> pushTemp: 1  
35 <E0> send: plus:  
36 <7C> returnTop
```



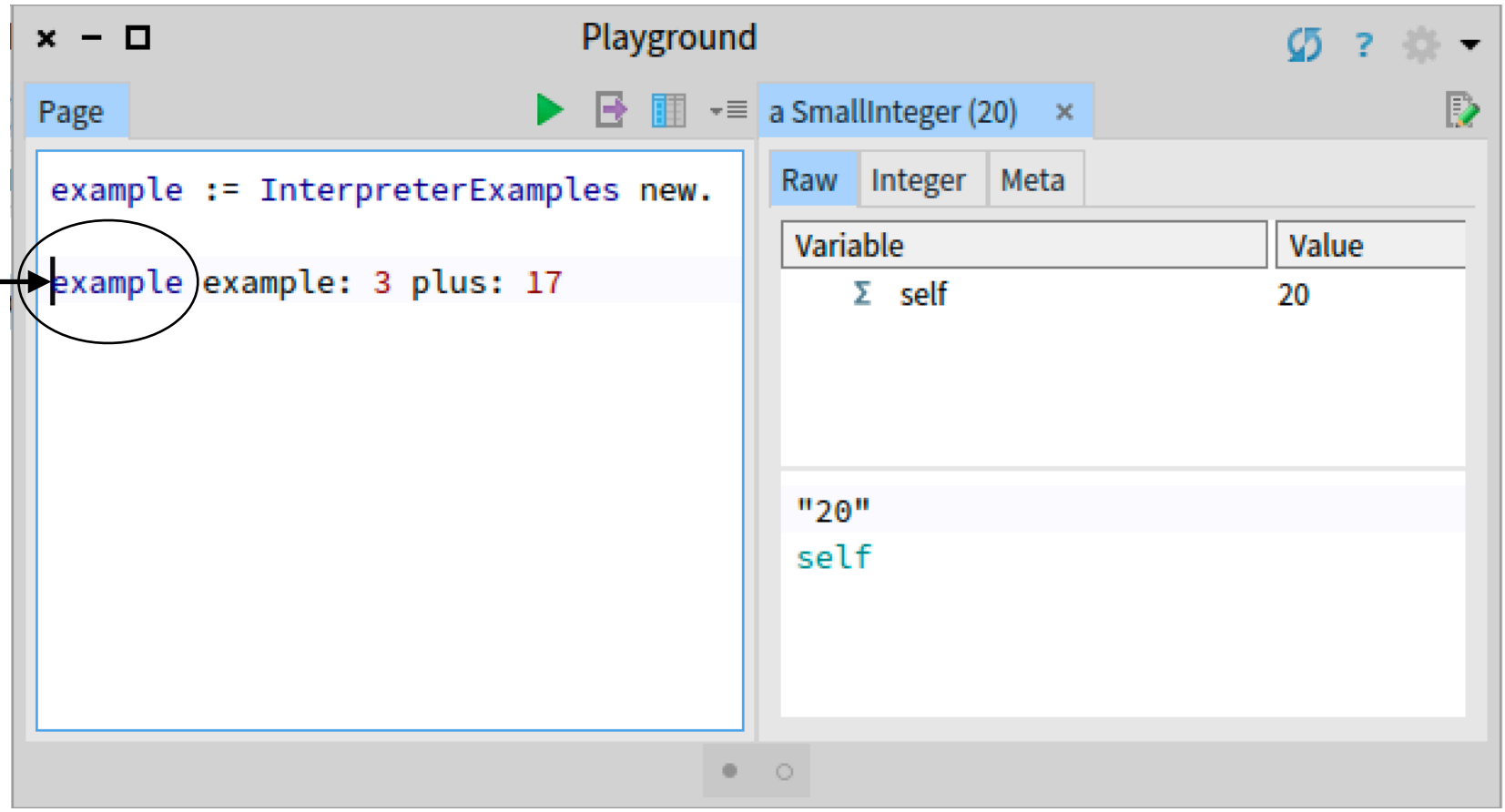
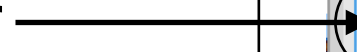
**Context mapping,  
Block,  
number of arguments**



**The context or nil**

...  
3  
17

Caller  
method1  
Flags  
context  
receiver



The screenshot shows a Scala Playground window titled "Playground". The code editor contains the following code:

```
example := InterpreterExamples new.  
example example: 3 plus: 17
```

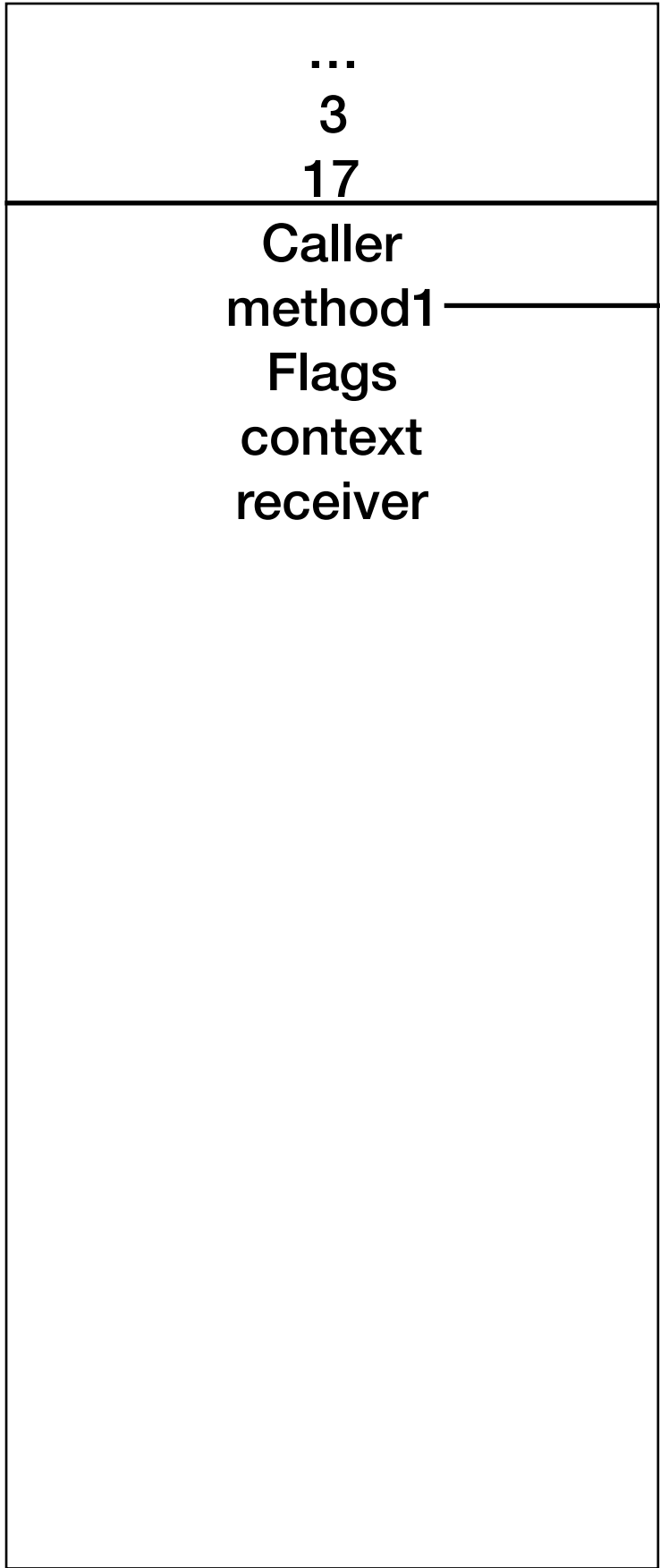
The second line of code is highlighted in light blue. A circle is drawn around the word "example" at the start of this line. An arrow from the "receiver" label in the diagram on the left points to this "example" word.

On the right side of the playground, there is a variable inspection panel for "a SmallInteger (20)". It has tabs for "Raw", "Integer", and "Meta". The "Raw" tab is selected. The panel contains a table with two columns: "Variable" and "Value".

Variable	Value
$\Sigma$ self	20

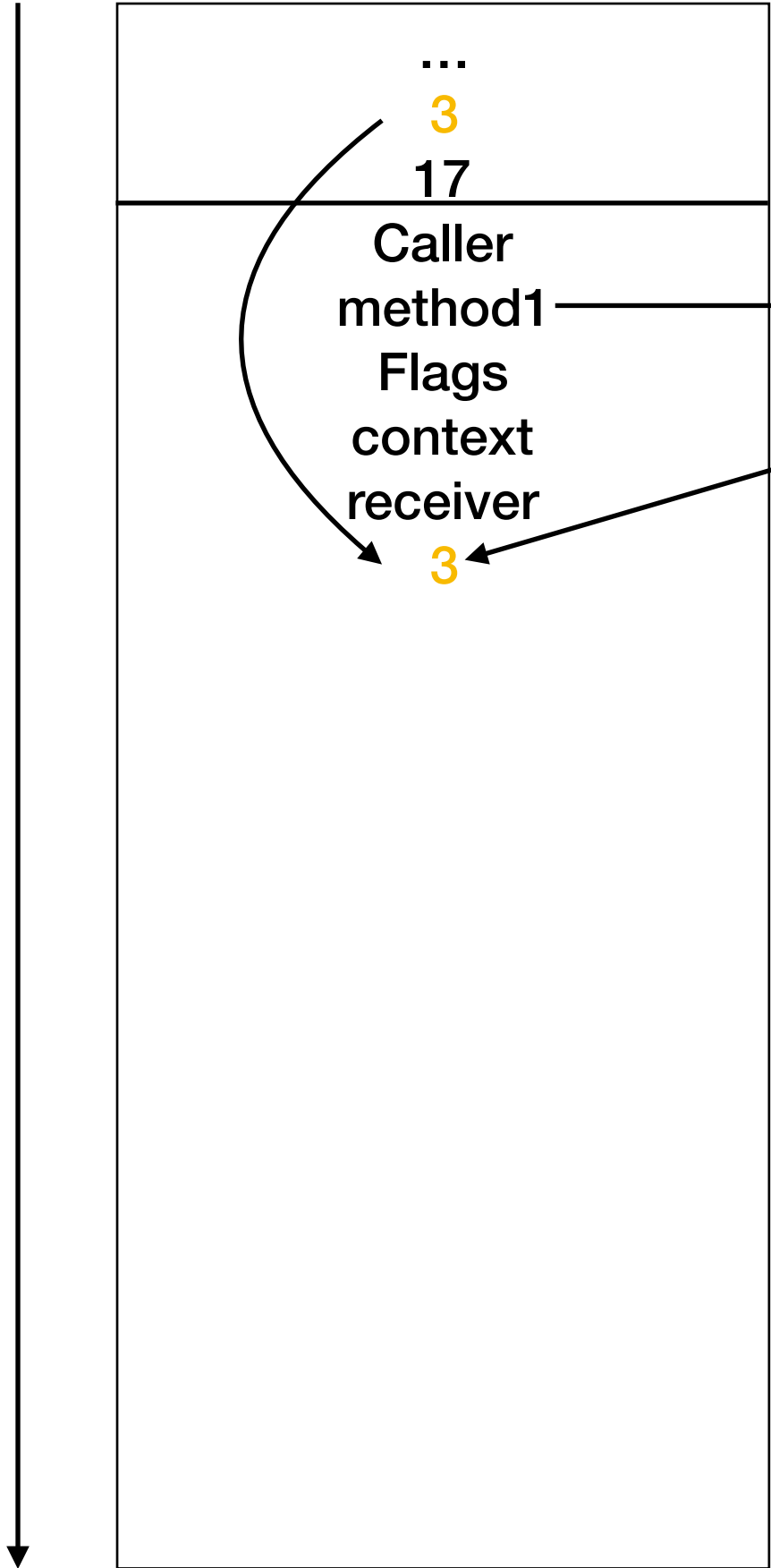
Below the table, the string representation of the value is shown as "20" and the object's class as self.



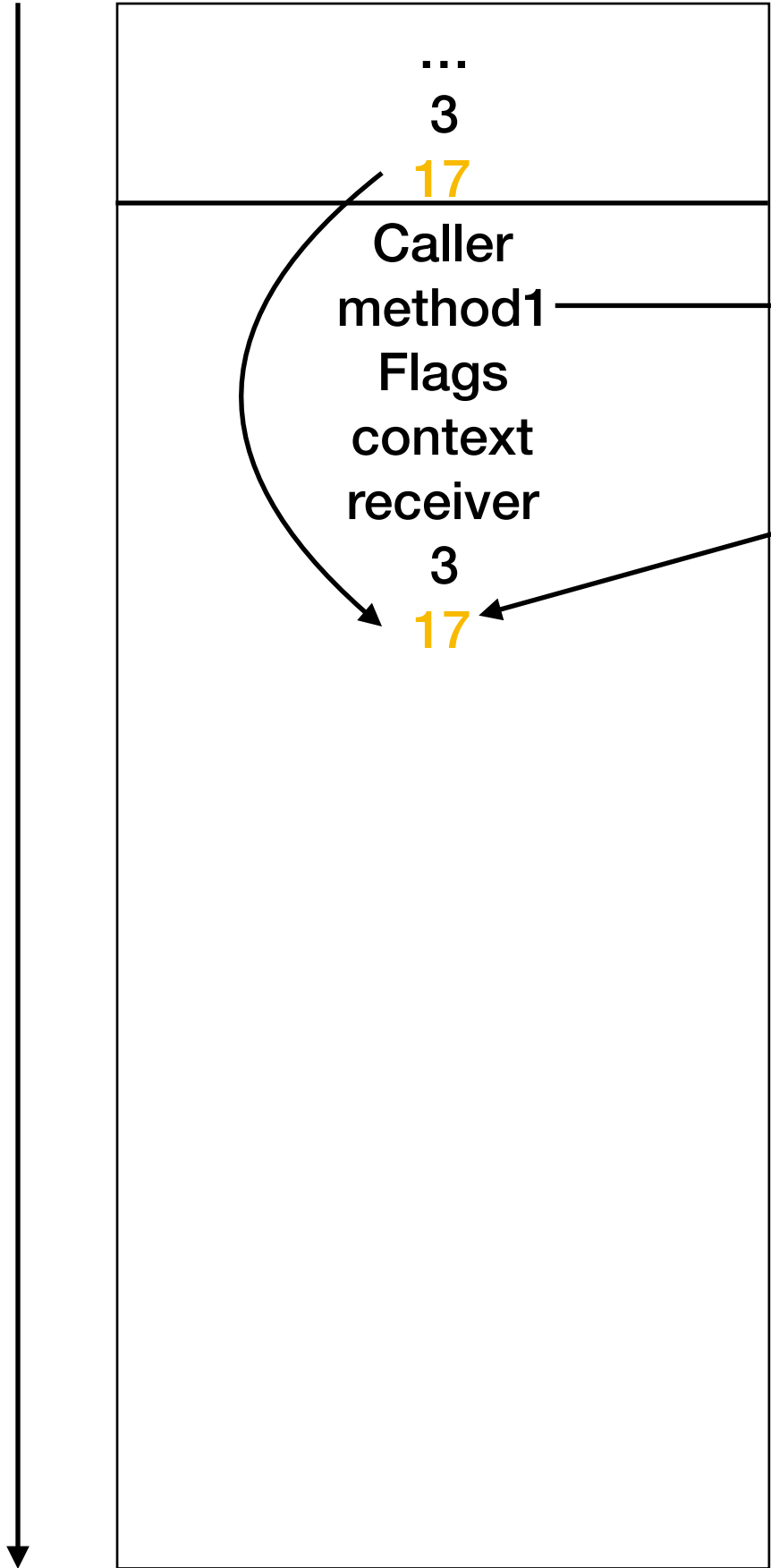


```
Inspector on a CompiledMethod (InterpreterExamples>>#e
a CompiledMethod (InterpreterExamples>>#example:plus:)
Raw Source Bytecode Ir AST Header Meta
33 <10> pushTemp: 0
34 <11> pushTemp: 1
35 <E0> send: plus:
36 <7C> returnTop
```

```
pushTemporaryV: x
pushTemporaryVariable: temporaryIndex
self internalPush: (self temporary: temporaryIndex in: localFP).
```

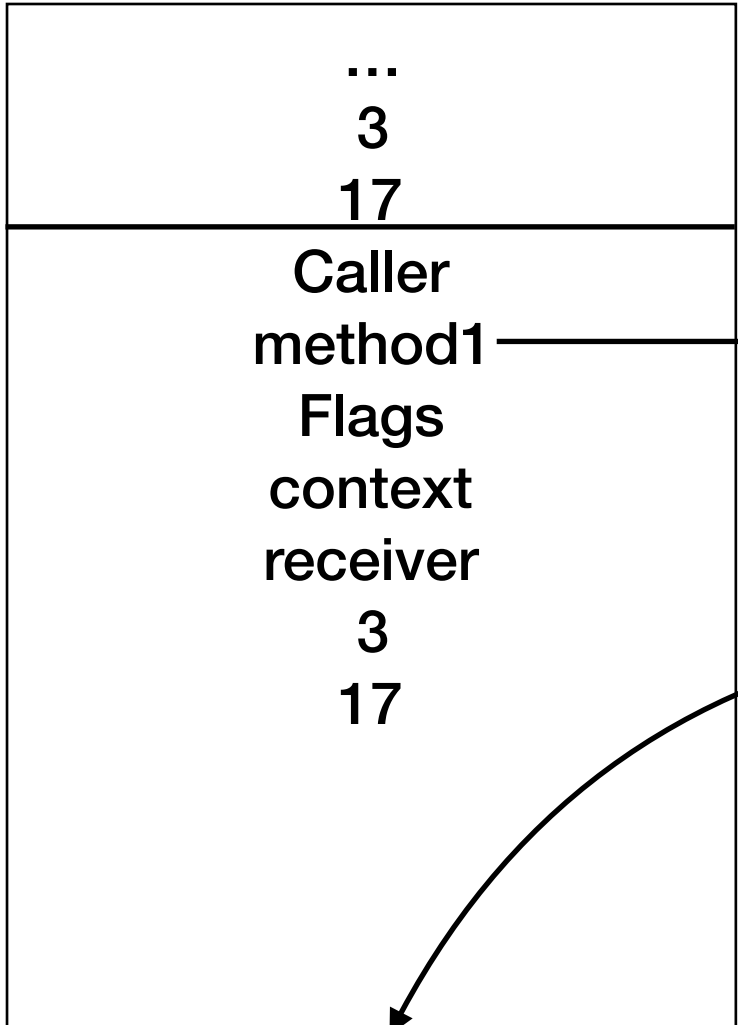


```
Inspector on a CompiledMethod (InterpreterExamples>>#e
a CompiledMethod (InterpreterExamples>>#example:plus:)
Raw Source Bytecode Ir AST Header Meta
33 <10> pushTemp: 0
34 <11> pushTemp: 1
35 <E0> send: plus:
36 <7C> returnTop
```



The screenshot shows a debugger window titled "Inspector on a CompiledMethod (InterpreterExamples>>#e". The selected object is "a CompiledMethod (InterpreterExamples>>#example:plus:)", and the "Bytecode" tab is active. The bytecode instructions are as follows:

```
33 <10> pushTemp: 0
34 <11> pushTemp: 1
35 <E0> send: plus:
36 <7C> returnTop
```

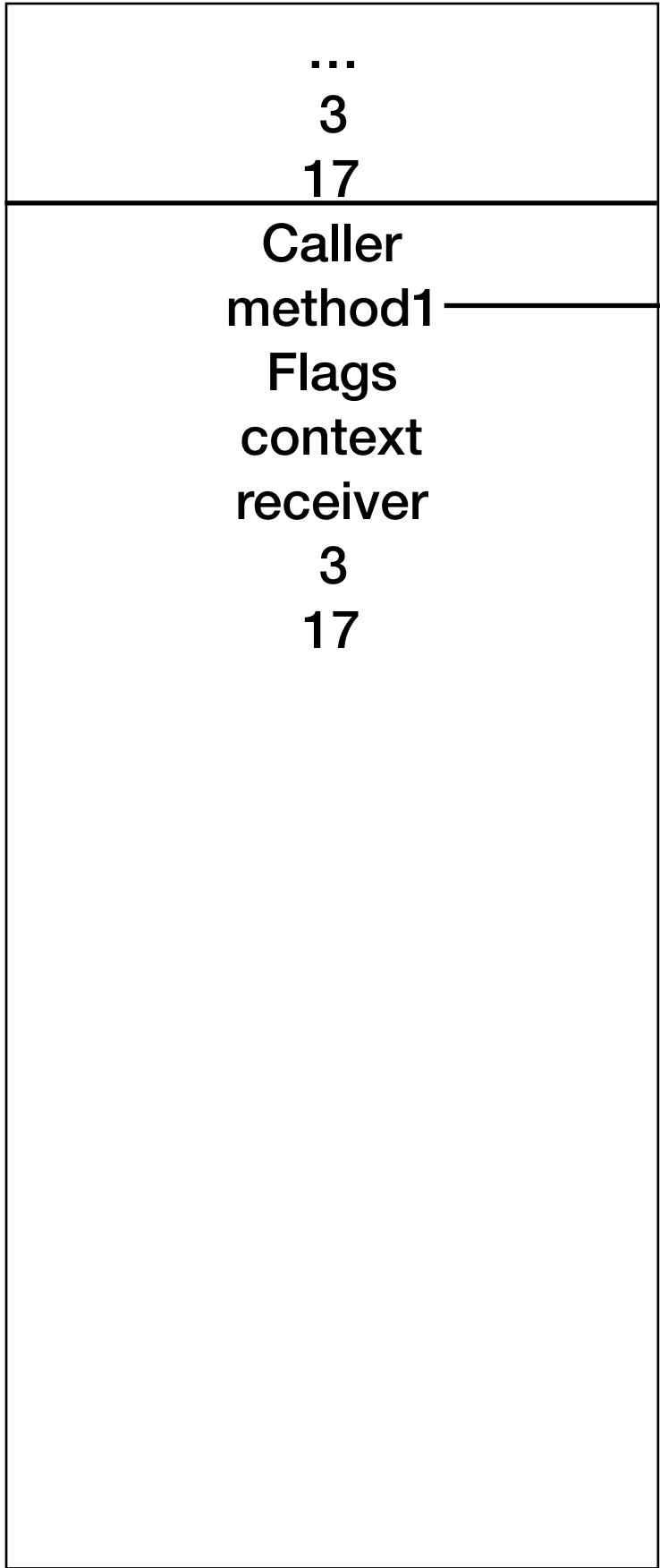


```

Inspector on a CompiledMethod (InterpreterExamples>>#e
a CompiledMethod (InterpreterExamples>>#example:plus:)
Raw Source Bytecode Ir AST Header Meta
33 <10> pushTemp: 0
34 <11> pushTemp: 1
35 <E0> send: plus:
36 <7C> returnTop
  
```

```

commonSendOrd x
commonSendOrdinary
"Send a message, starting lookup with the receiver's class."
"Assume: messageSelector and argumentCount have been set, and that
the receiver and arguments have been pushed onto the stack,"
"Note: This method is inlined into the interpreter dispatch loop."
<sharedCodeInCase: #singleExtendedSendBytecode>
self sendBreakpoint: messageSelector receiver: (self internalStackValue: argumentCount).
self printSends ifTrue:
    [self printActivationNameForSelector: messageSelector startClass: (objectMemory classForClassTag:
lookupClassTag); cr].
self internalFindNewMethodOrdinary.
self internalExecuteNewMethod.
self fetchNextBytecode
  
```



Inspector on a CompiledMethod (InterpreterExamples>>#e)

a CompiledMethod (InterpreterExamples>>#example:plus:)

Raw Source **Bytecode** Ir AST Header Meta

```

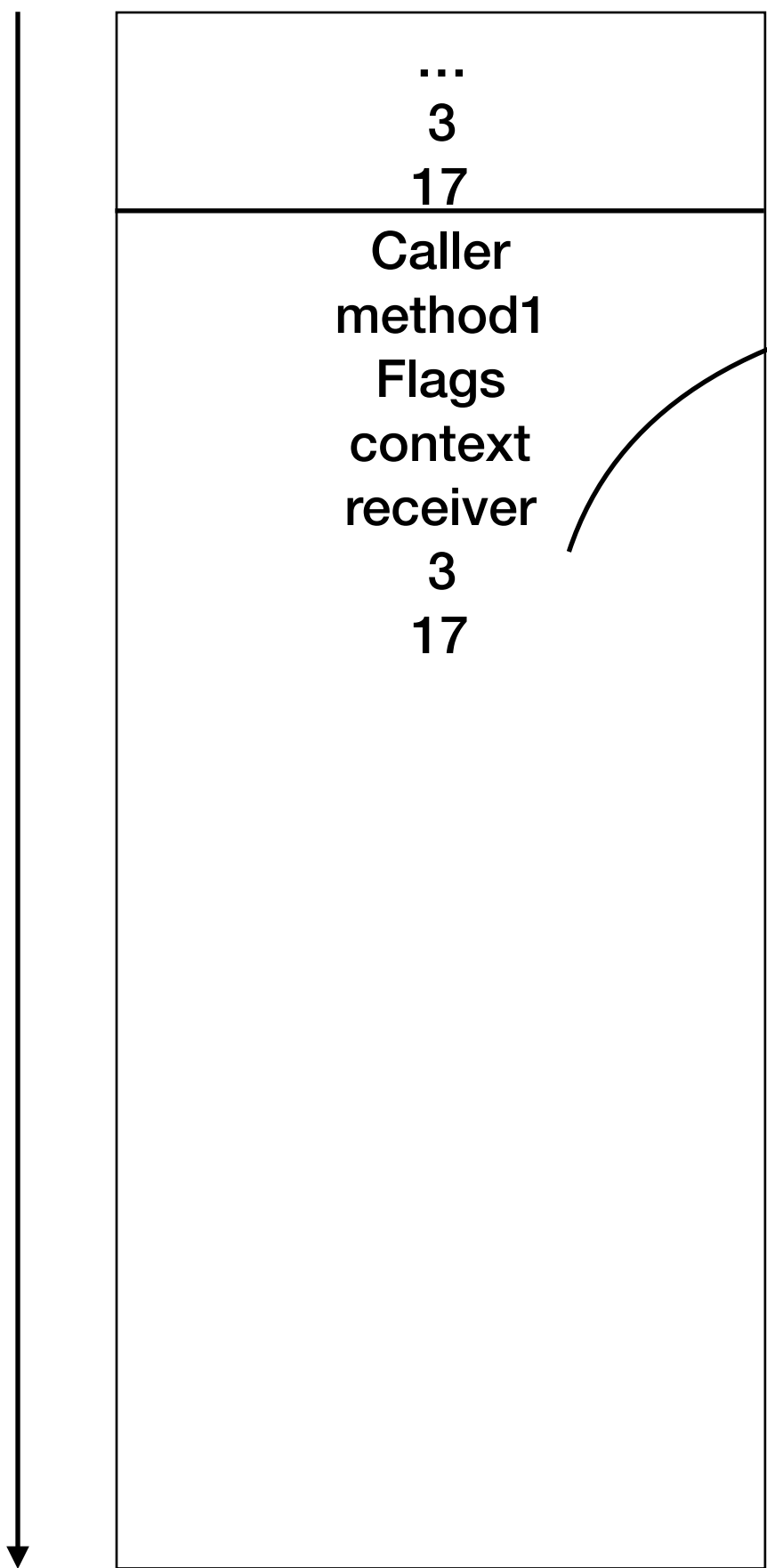
33 <10> pushTemp: 0
34 <11> pushTemp: 1
35 <E0> send plus:
36 <7C> returnTop
  
```

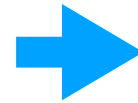
Inspector on a CompiledMethod (InterpreterExample

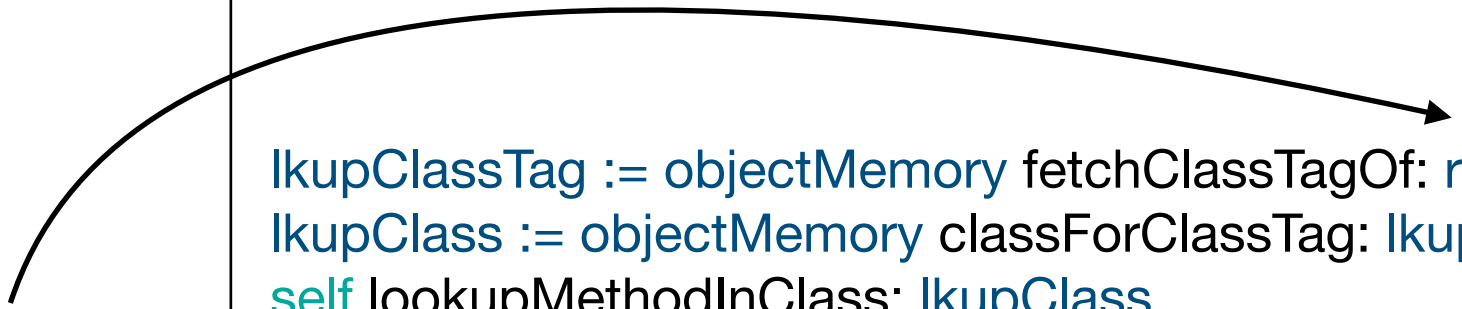
a CompiledMethod (InterpreterExamples>>#example:plus:)

Raw Source Bytecode Ir AST Header Meta

Variable	Value
{ } self	InterpreterExamples>>#example:plus:
▶ ¶ literal1	#plus:
▶ Σ bc 33	16
▶ Σ bc 34	17
▶ Σ bc 35	224
▶ Σ bc 36	124
▶ Σ bc 37	211
▶ Σ bc 38	7
▶ Σ bc 39	0
▶ Σ bc 40	253



 `self internalFindNewMethodOrdinary.`  
`self internalExecuteNewMethod.`  
`self fetchNextBytecode`

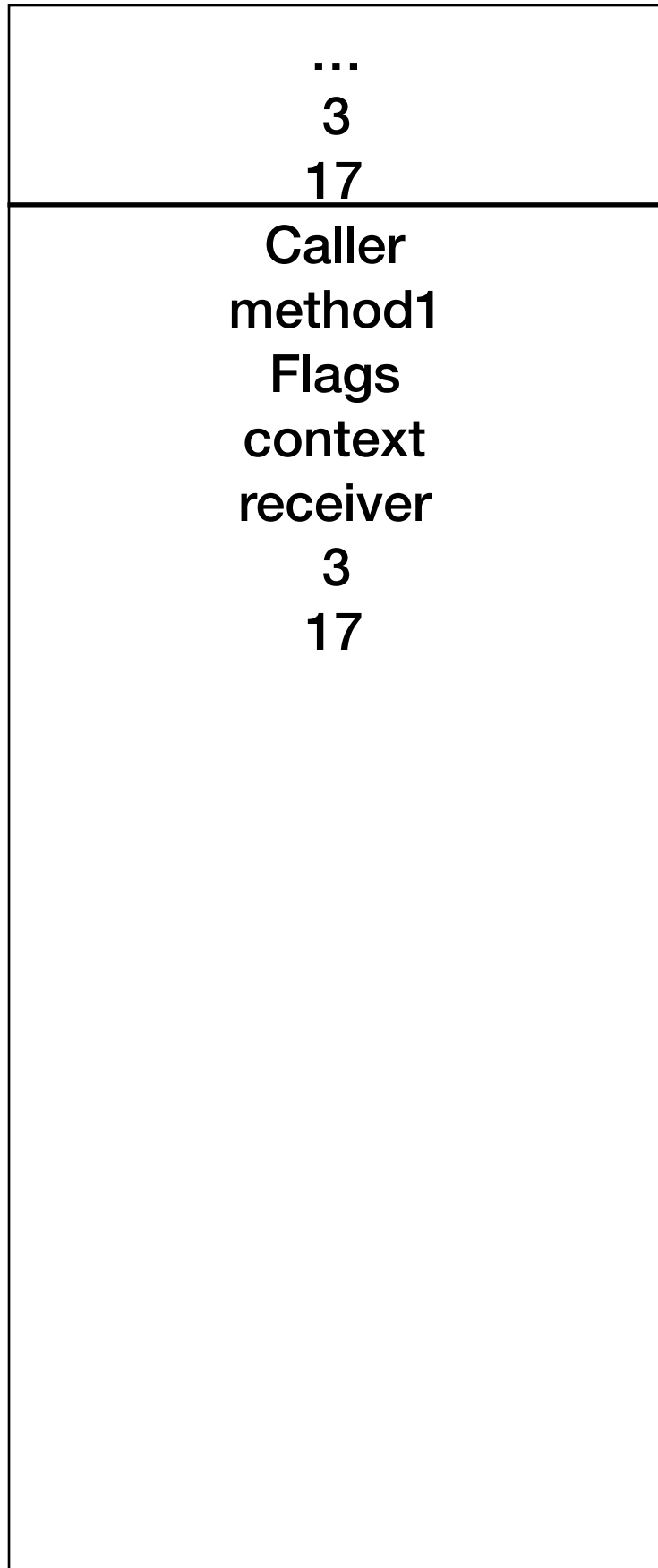


`IkupClassTag := objectMemory fetchClassTagOf: rcvr.`  
`IkupClass := objectMemory classForClassTag: IkupClassTag.`  
`self lookupMethodInClass: IkupClass`



**SmallInteger**





**Case there is no method dictionary.**

**SmallInteger MethodDictionary**

**Instance Variable**

**(Selector**

\*

+

-

/

...

**array of Methods**

\* method

+ method

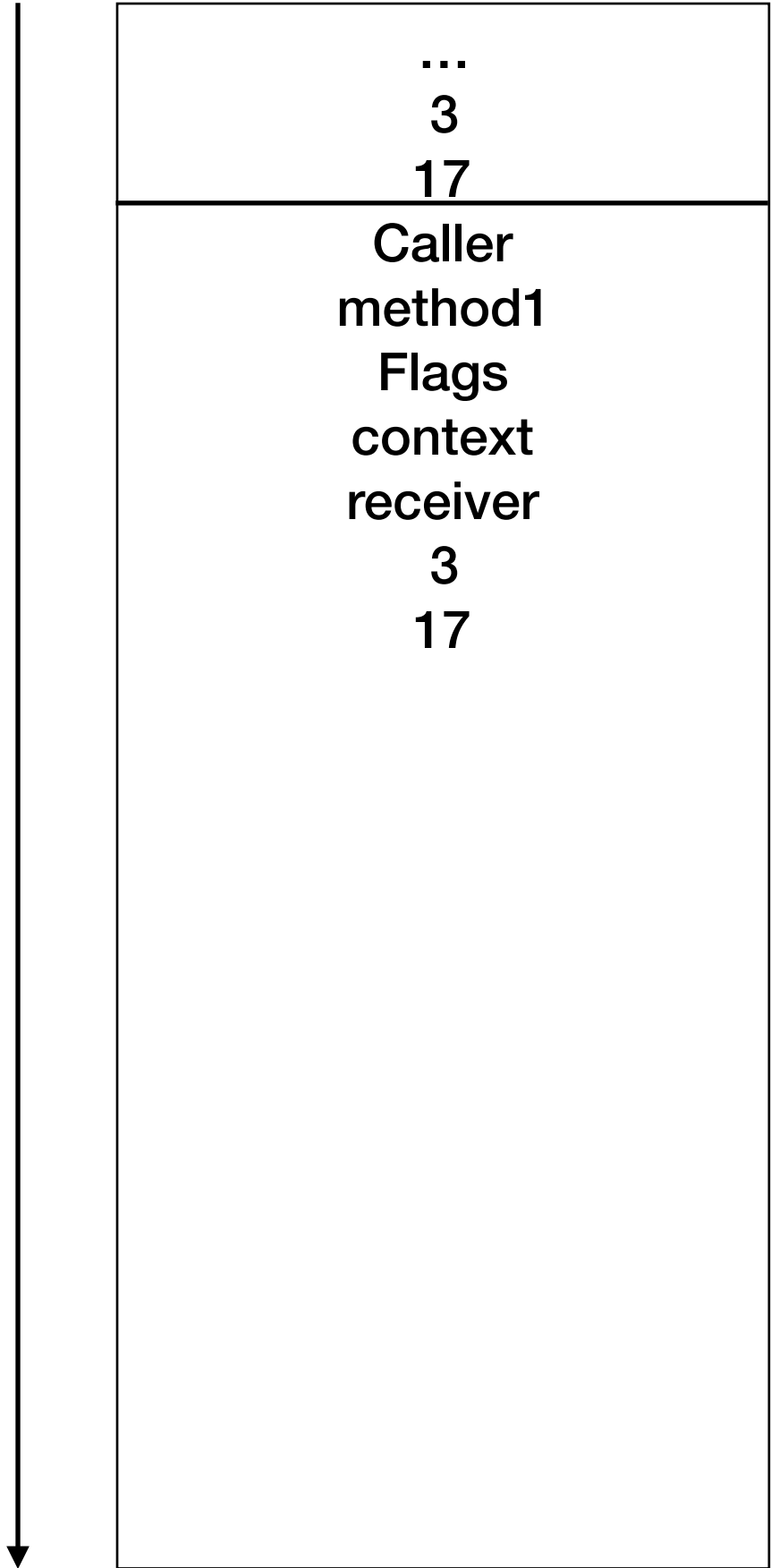
- method

method

...

**Send the message #CannotInterpret: to superclass**





```

self internalFindNewMethodOrdinary.
self internalExecuteNewMethod.
self fetchNextBytecode

```

**ProtoObject**

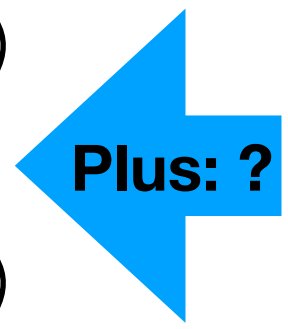
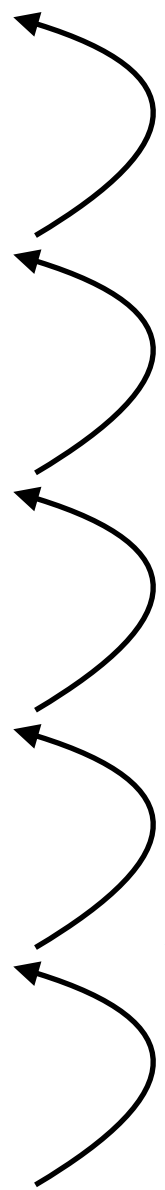
**Object**

**Magnitude**

**Number**

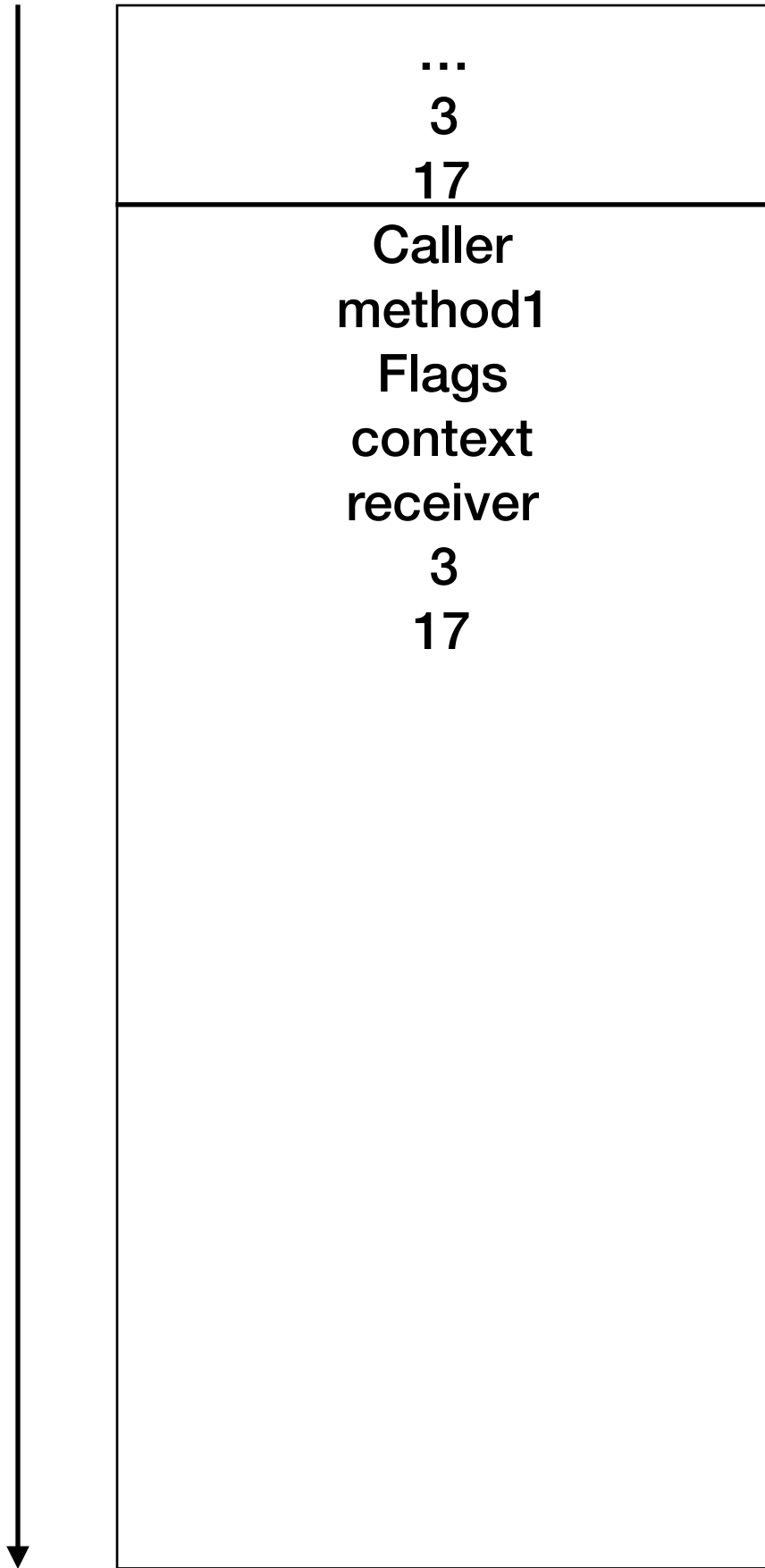
**Integer**

**SmallInteger**



**Plus: ?**

**Plus: Not found**



➔ `self internalFindNewMethodOrdinary.`  
`self internalExecuteNewMethod.`  
`self fetchNextBytecode`

**ProtoObject**

**Object**

**Magnitude**

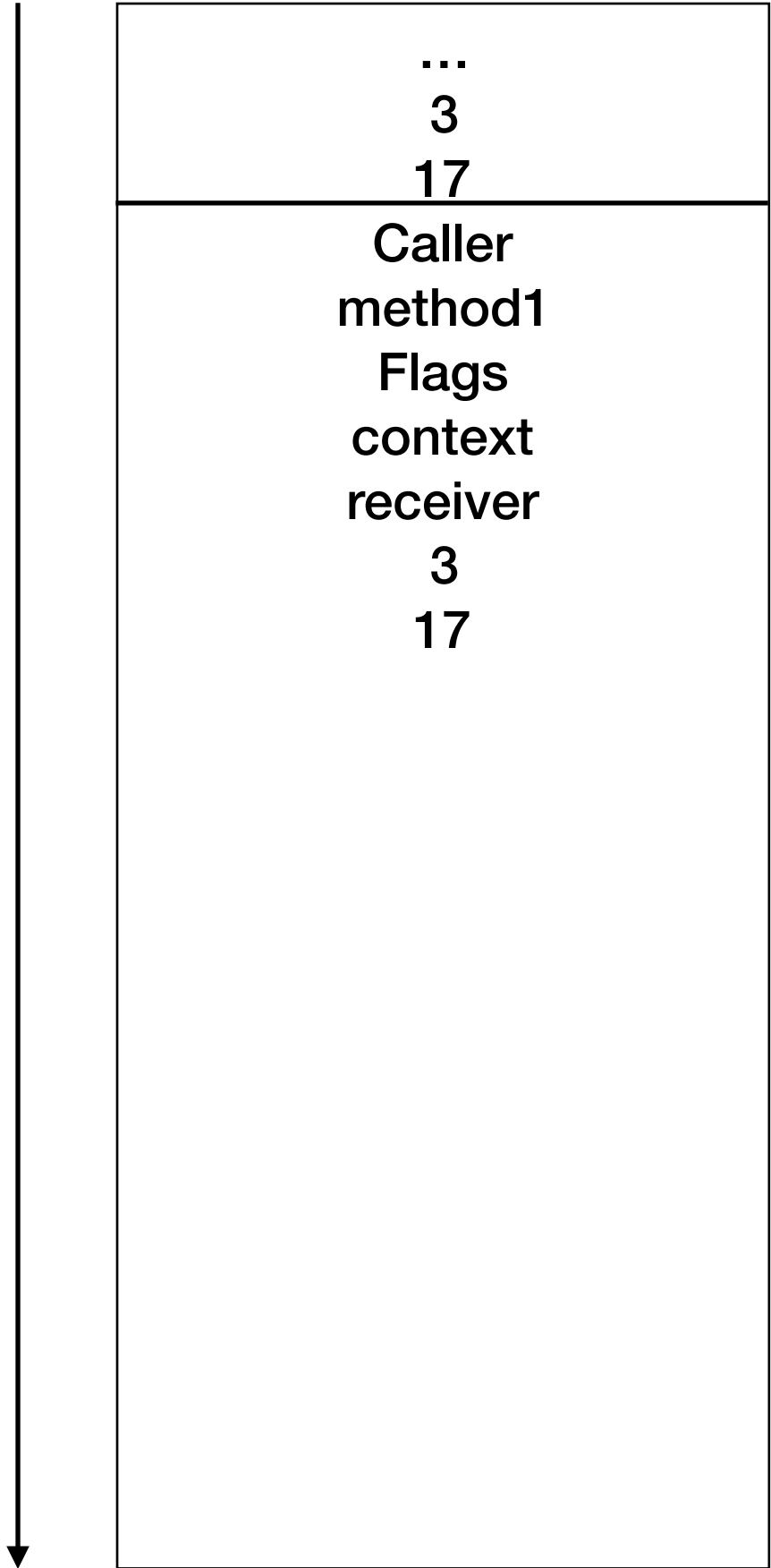
**Number**

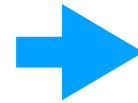
**Integer**

**SmallInteger**

**Plus: ?**

**Plus: Not found**





```

self internalFindNewMethodOrdinary.
self internalExecuteNewMethod.
self fetchNextBytecode

```

**ProtoObject**

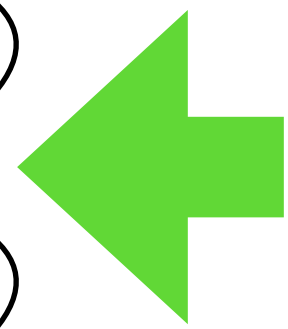
**Object**

**Magnitude**

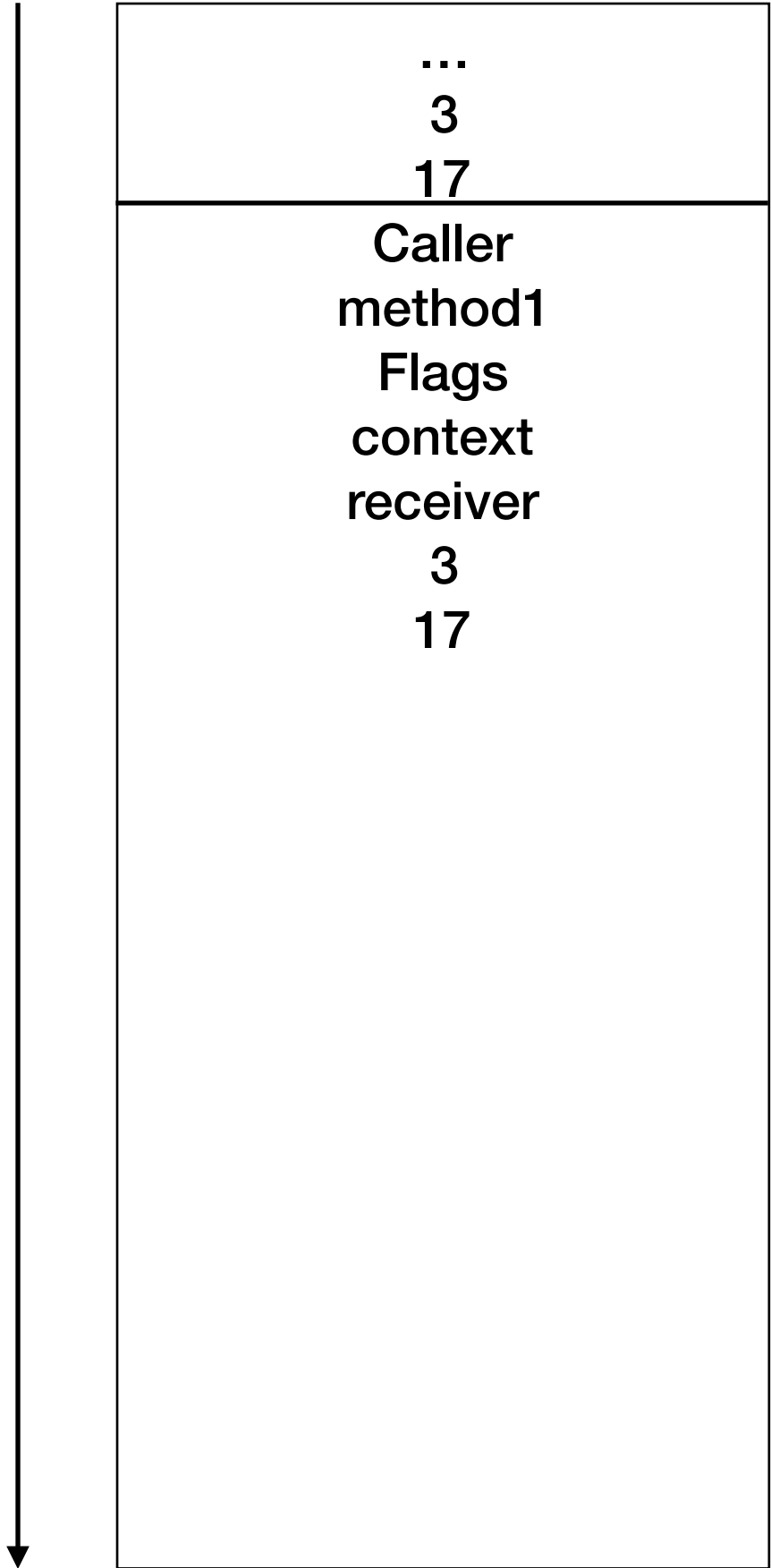
**Number**

**Integer**

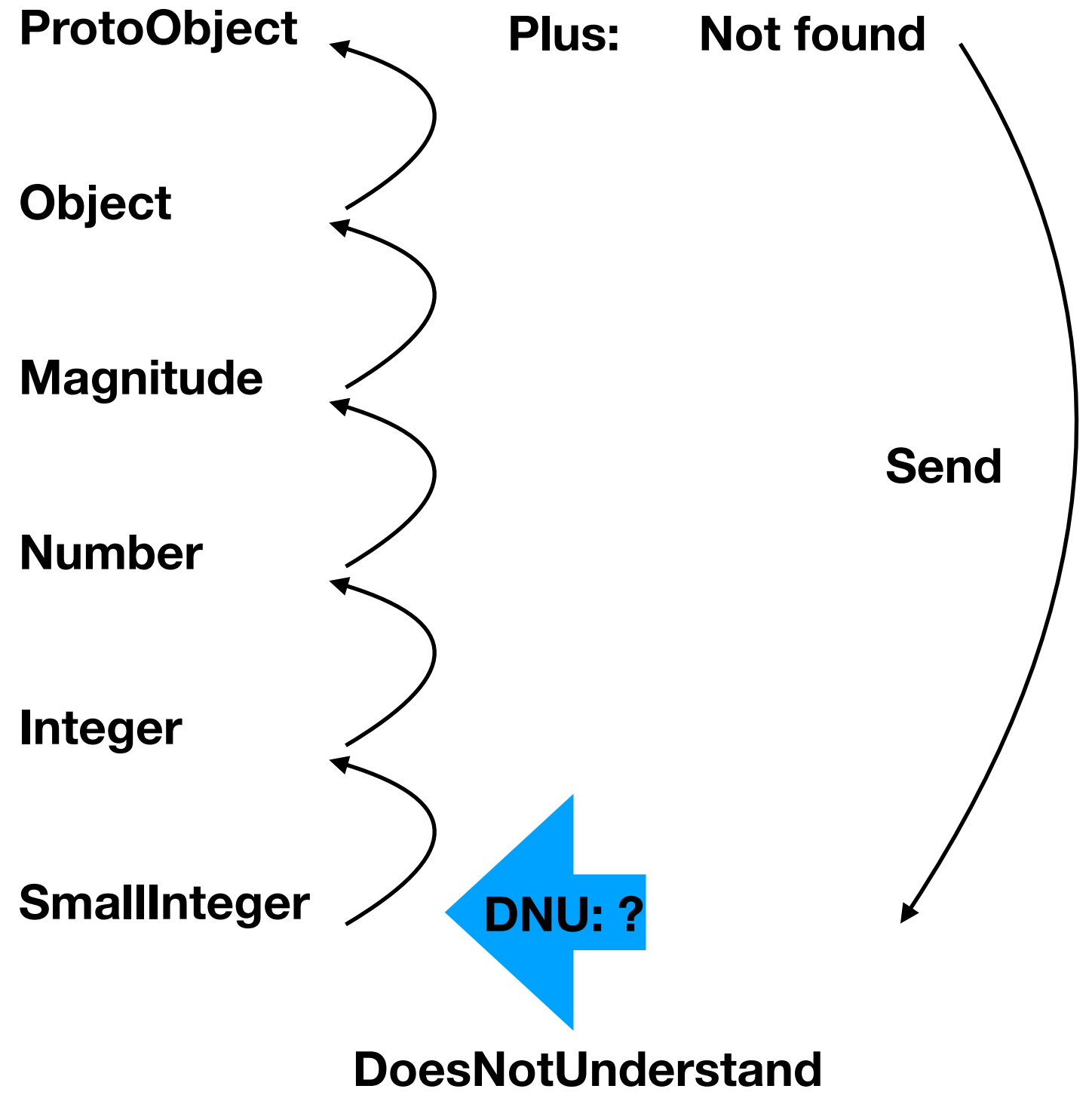
**SmallInteger**

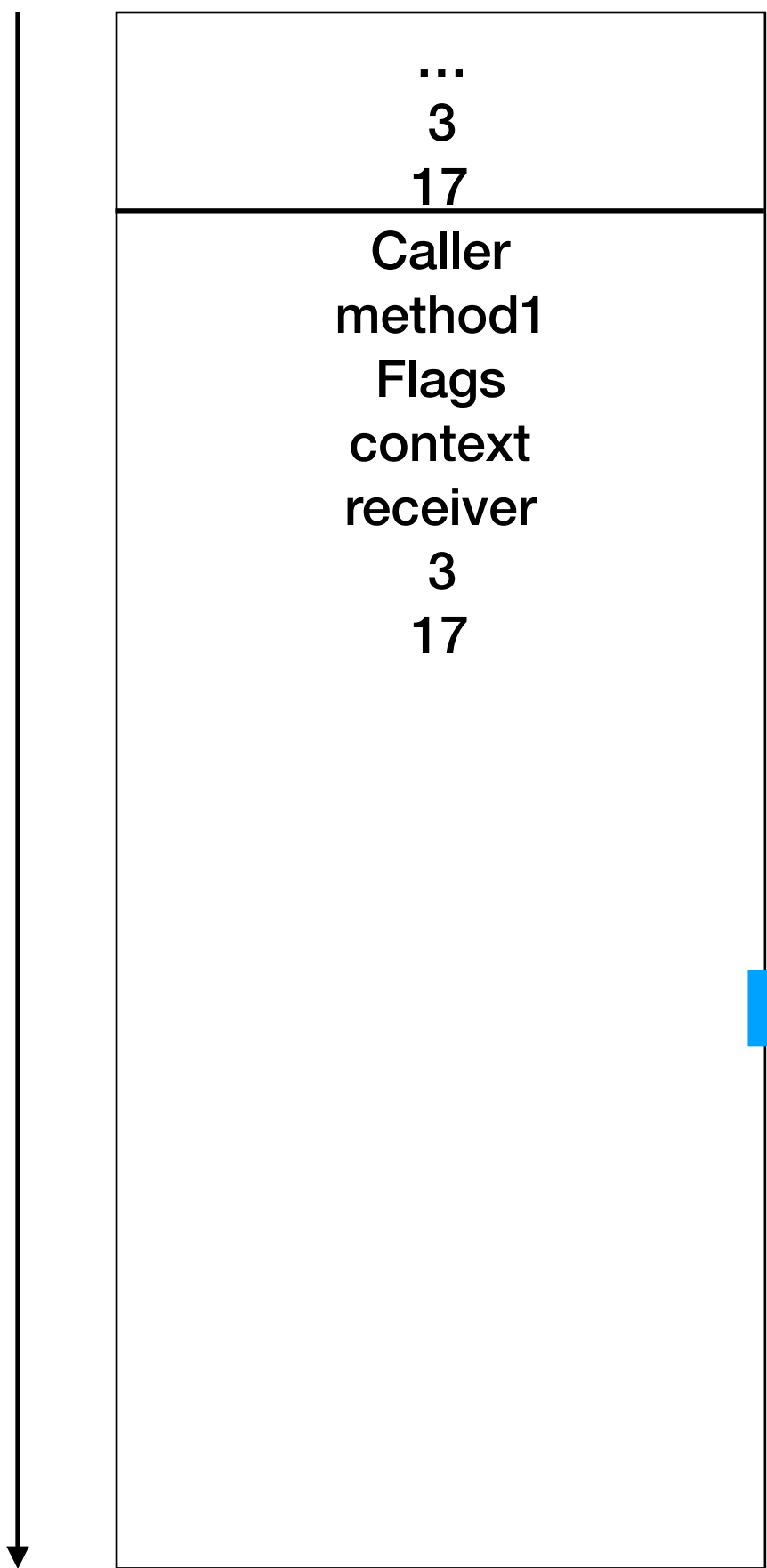


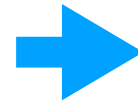
**#Plus: is installed here**



**Case #Plus: is nowhere to be found.**







```

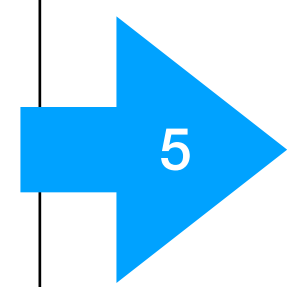
self internalFindNewMethodOrdinary.
self internalExecuteNewMethod.
self fetchNextBytecode

```

**Number**

**MethodDictionary**

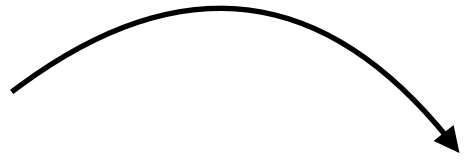
**Instance variable  
(Selectors)**



...

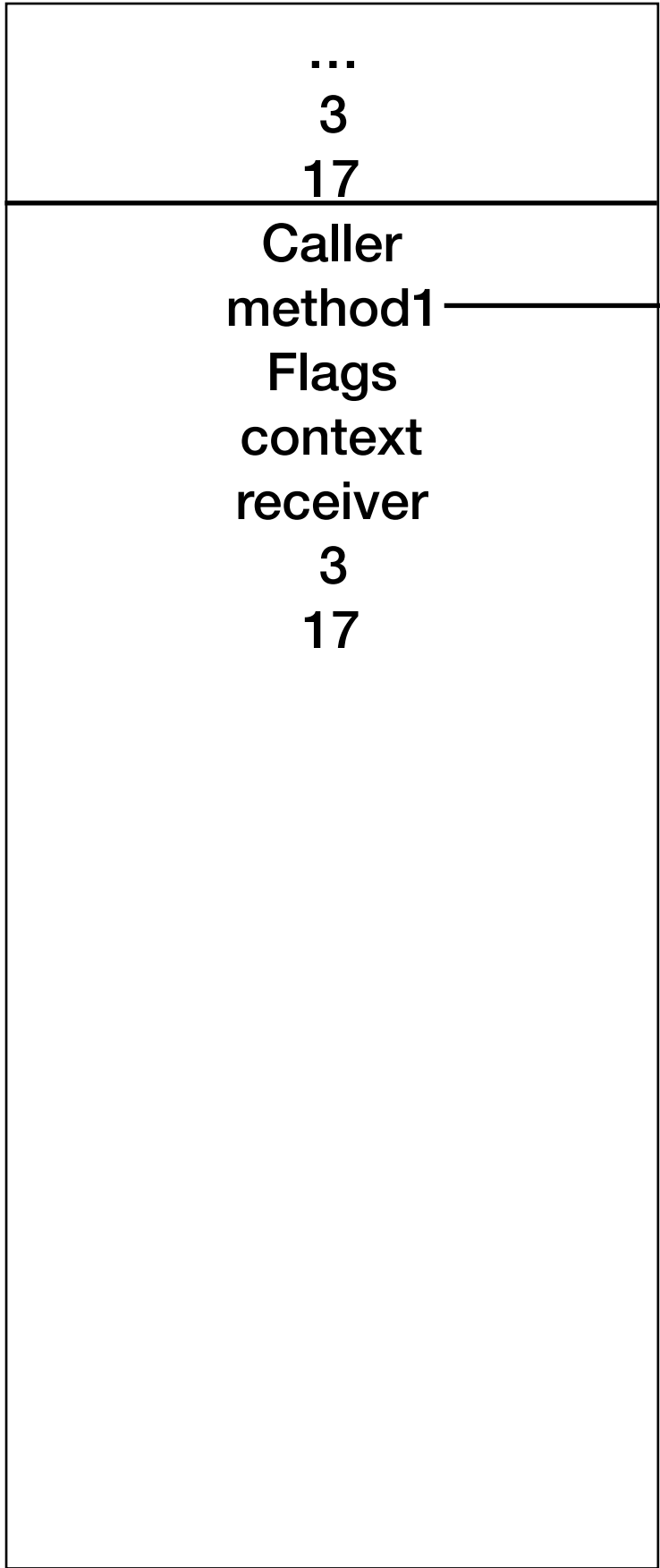
**Plus:**

...



**arrayOfMethods**

1
2
3
4
<b>Plus: method</b>
...



```

self internalFindNewMethodOrdinary.
self internalExecuteNewMethod.
self fetchNextBytecode

```

Inspector on a CompiledMethod (InterpreterExamples>>#e)

a CompiledMethod (InterpreterExamples>>#example:plus:)

Raw Source Bytecode Ir AST Header Meta

```

33 <10> pushTemp: 0
34 <11> pushTemp: 1
35 <E0> send: plus:
36 <7C> returnTop

```

Found

plus: x

```

plus: aNumber
^self + aNumber

```

Inspector on a CompiledMethod (Number>>#plus:)

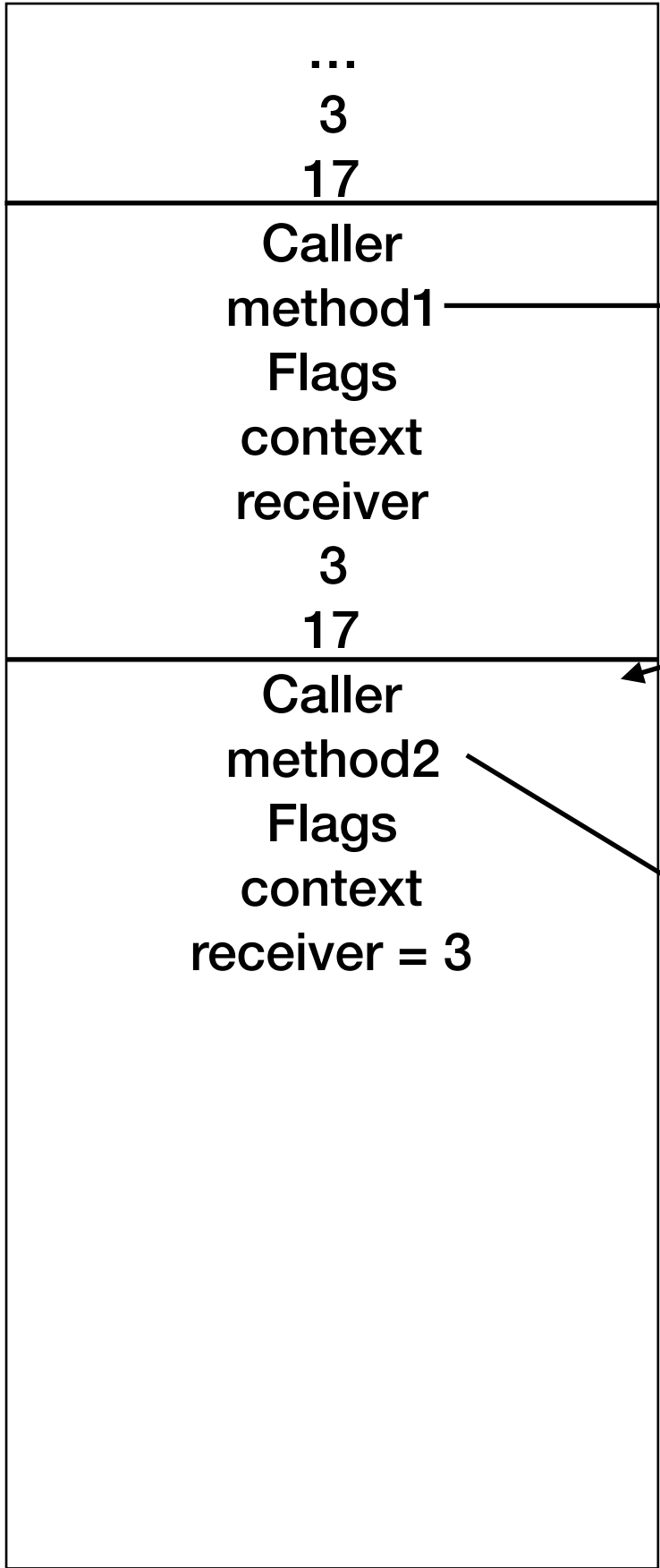
a CompiledMethod (Number>>#plus:)

Raw Source Bytecode Ir AST Header Meta

```

25 <70> self
26 <10> pushTemp: 0
27 <B0> send: +
28 <7C> returnTop

```



```

self internalFindNewMethodOrdinary.
self internalExecuteNewMethod.
self fetchNextBytecode

```

Inspector on a CompiledMethod (InterpreterExamples>>#e)

a CompiledMethod (InterpreterExamples>>#example:plus:)

Raw Source Bytecode Ir AST Header Meta

```

33 <10> pushTemp: 0
34 <11> pushTemp: 1
35 <E0> send: plus:
36 <7C> returnTop

```

plus: x

```

plus: aNumber
^self + aNumber

```

Inspector on a CompiledMethod (Number>>#plus:)

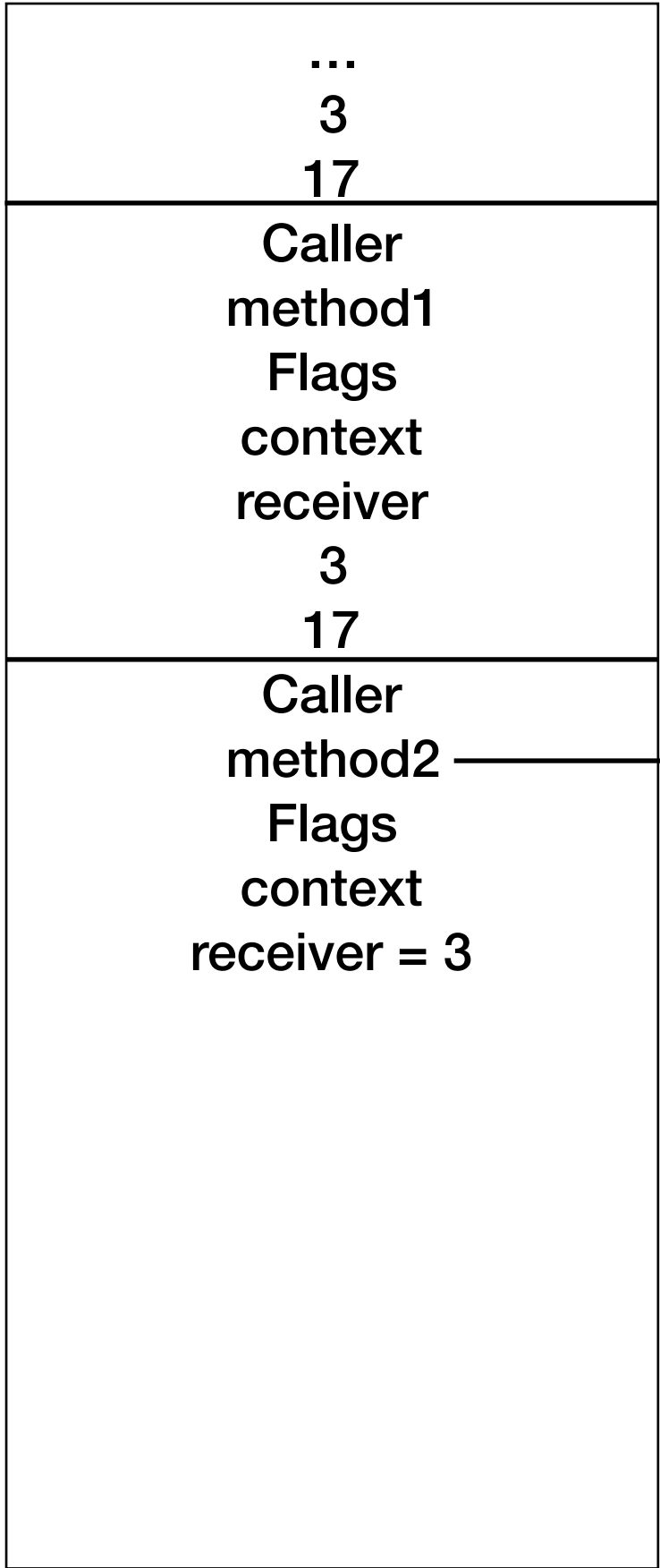
a CompiledMethod (Number>>#plus:)

Raw Source Bytecode Ir AST Header Meta

```

25 <70> self
26 <10> pushTemp: 0
27 <B0> send: +
28 <7C> returnTop

```



```

self internalFindNewMethodOrdinary.
self internalExecuteNewMethod.
self fetchNextBytecode

```

plus: x

```

plus: aNumber
  ^self + aNumber

```

Inspector on a CompiledMethod (Number>>#plus:)

a CompiledMethod (Number>>#plus:)

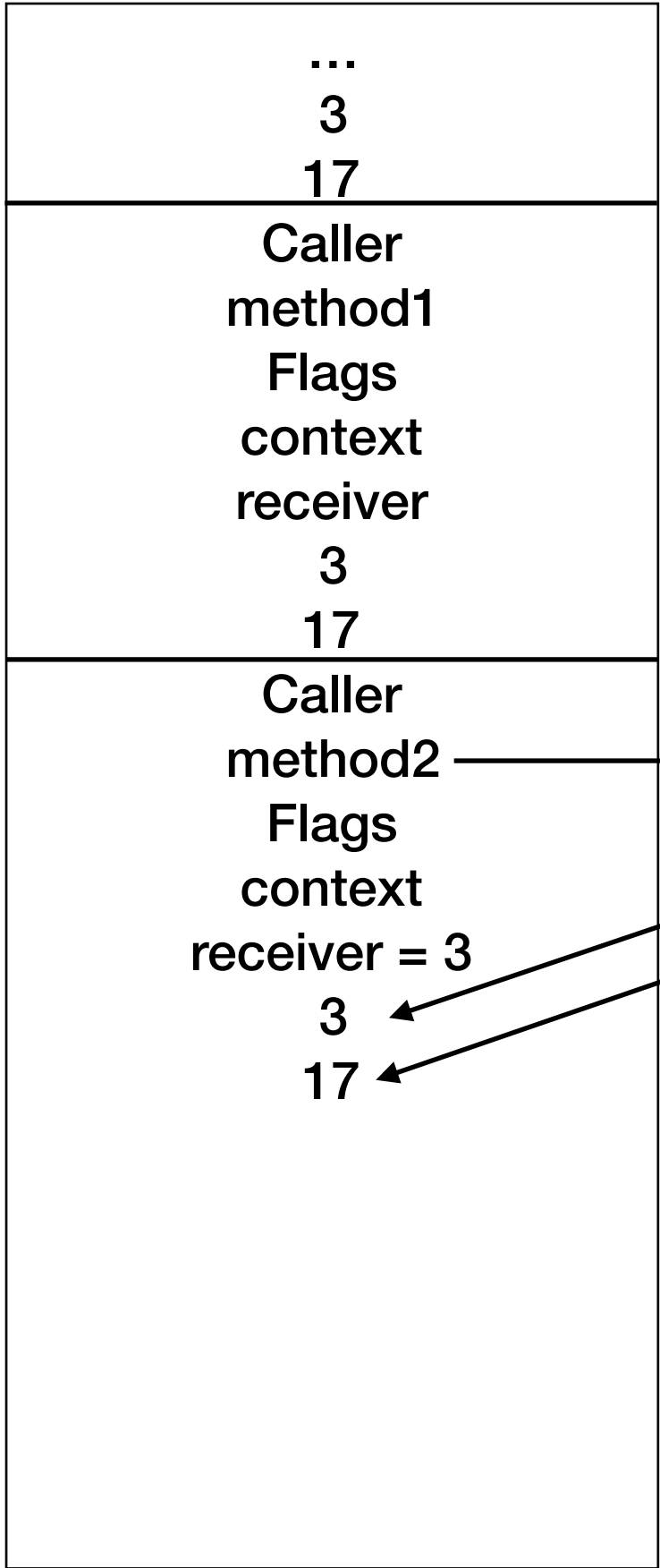
Raw Source **Bytecode** Ir AST Header Meta

```

25 <70> self
26 <10> pushTemp: 0
27 <B0> send: +
28 <7C> returnTop

```





```
plus: aNumber  
  ^self + aNumber
```

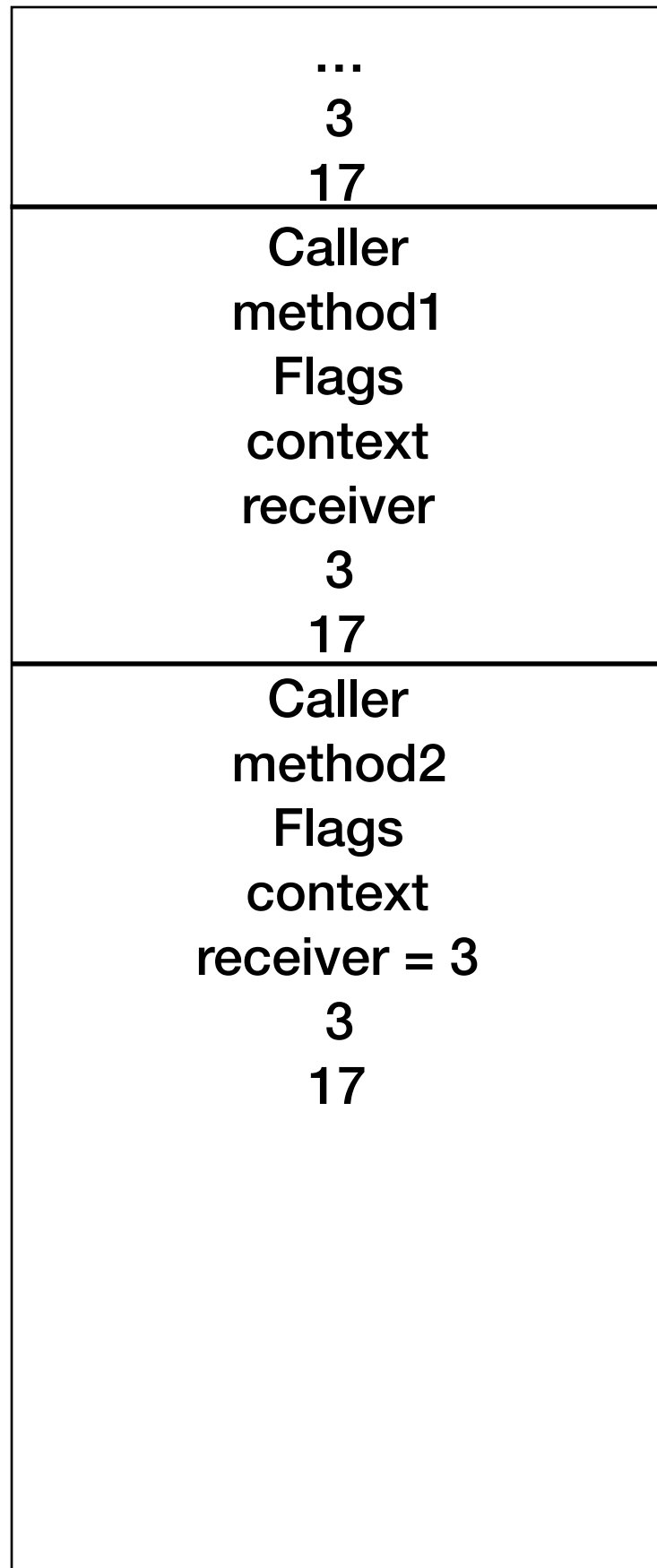
Inspector on a CompiledMethod (Number>>#plus:)

a CompiledMethod (Number>>#plus:)

Raw Source Bytecode Ir AST Header Meta

```
25 <70> self  
26 <10> pushTemp: 0  
27 <B0> send: +  
28 <7C> returnTop
```

# Debugger view



debug it

Stack

Proceed Restart Into Over Through

Bytecode

SmallInteger(Number) plus:  
InterpreterExamples example:plus:  
UndefinedObject Dolt

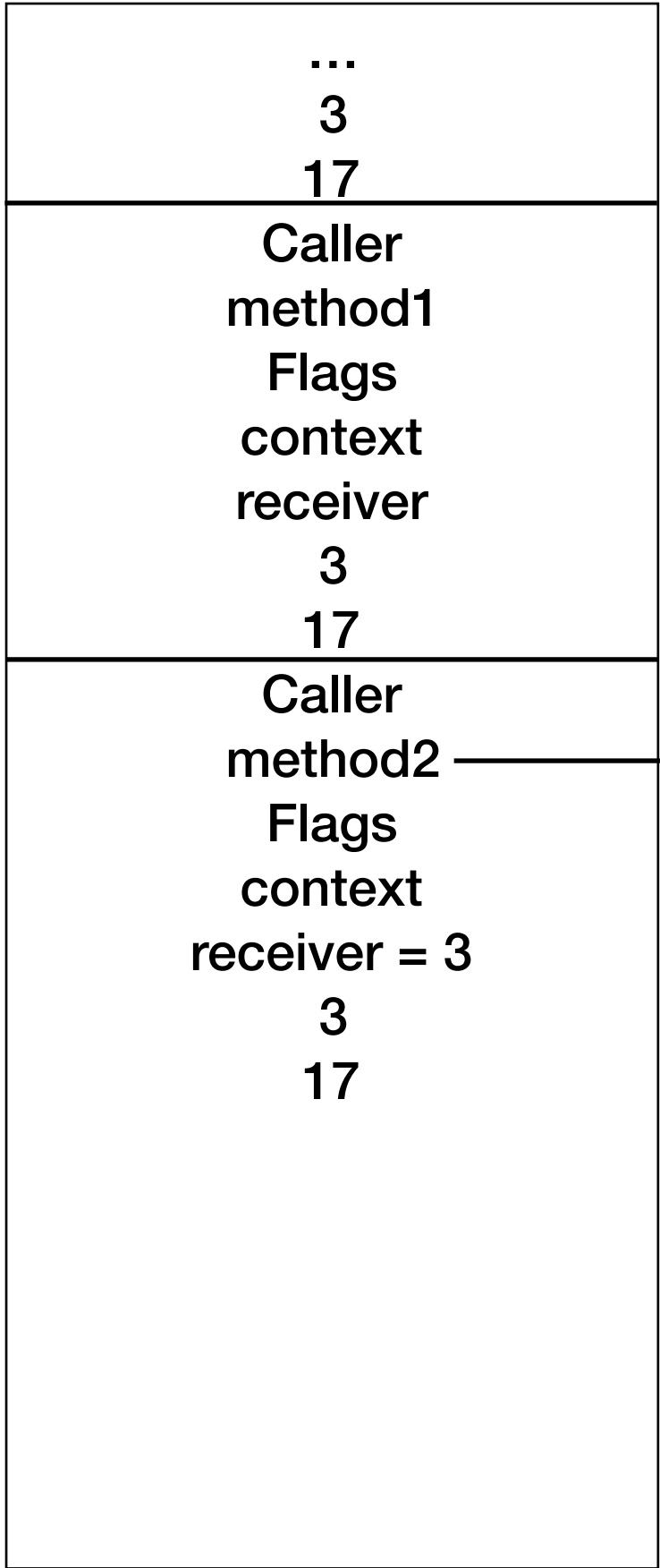
Source

Where is? Browse

```
plus: aNumber  
  ^self + aNumber
```

Variables Evaluator

Type	Variable	Value
implicit	self	3
parameter	aNumber	17
implicit	thisContext	SmallInteger(Number)>>plus:



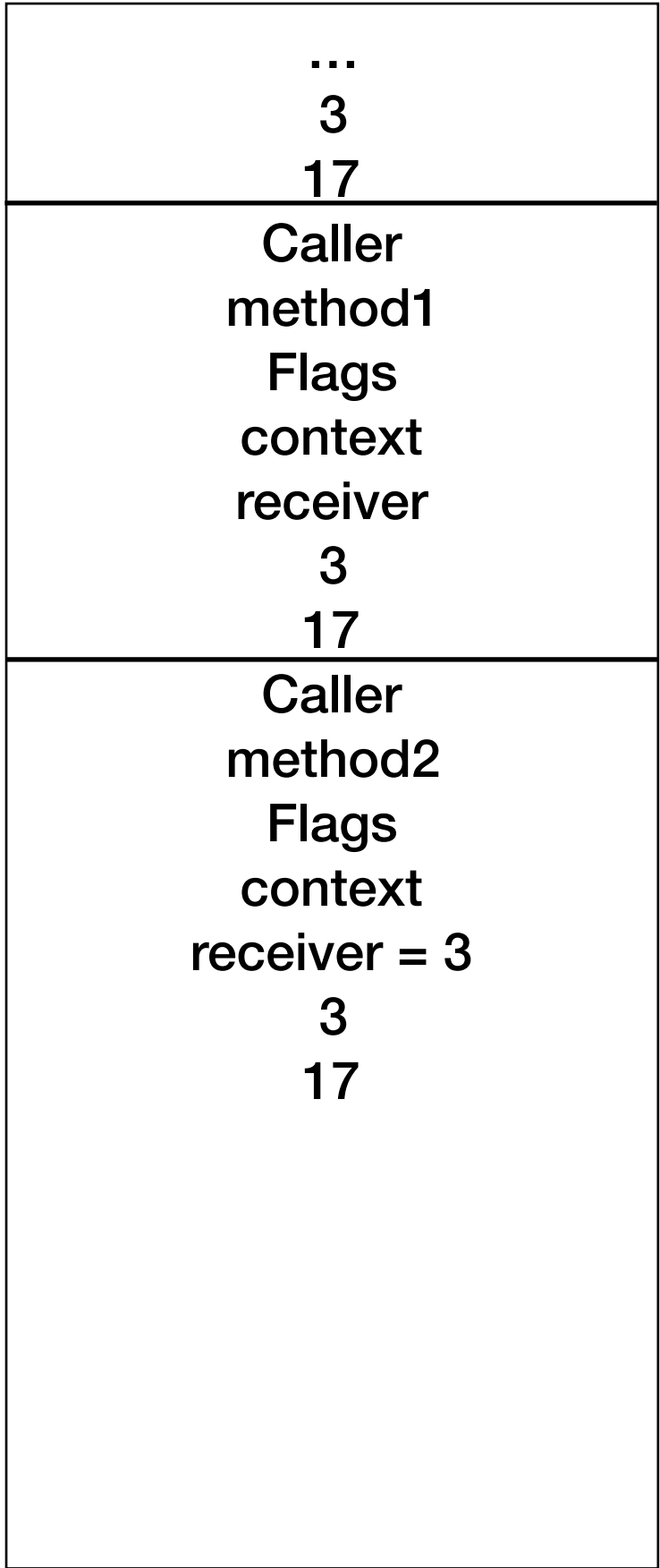
plus: aNumber  
^self + aNumber

Inspector on a CompiledMethod (Number>>#plus:)

a CompiledMethod (Number>>#plus:)

Raw Source **Bytecode** Ir AST Header Meta

25 <70> self  
26 <10> pushTemp: 0  
2 <B0> send: +  
28 <7C> returnTop



```
+ aNumber
  "Primitive. Add the receiver to the argument and answer with the result
  if it is a SmallInteger. Fail if the argument or the result is not a
  SmallInteger Essential No Lookup. See Object documentation whatIsAPrimitive."

  <primitive: 1>
  ^ super + aNumber
```

Inspector on a CompiledMethod (SmallInteger>>#+)

a CompiledMethod (SmallInteger>>#+)

Raw Source Pragas Bytecode Ir AST Header Meta

```
33 <8B 01 00> callPrimitive: 1
36 <70> self
37 <10> pushTemp: 0
38 <85 20> superSend: +
40 <7C> returnTop
```

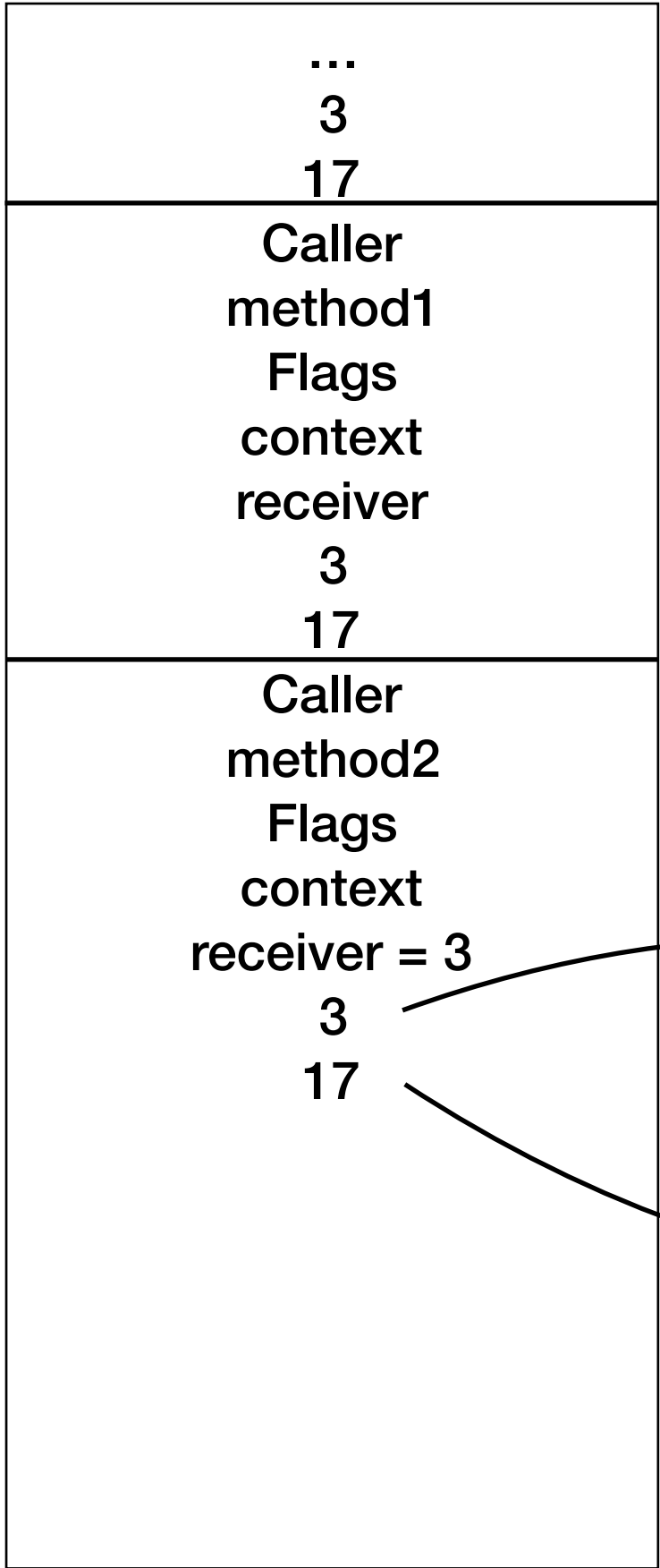
primitiveAdd

```
self pop2AndPushIntegerIfOK: (self stackIntegerValue: 1) + (self stackIntegerValue: 0)
```

1/3 [1] -- 1/3 [13] arithmetic integer primitives extension F +L W

[pop2AndPushIntegerIfOK:] Super and Self Messages sent but not implemented X ?

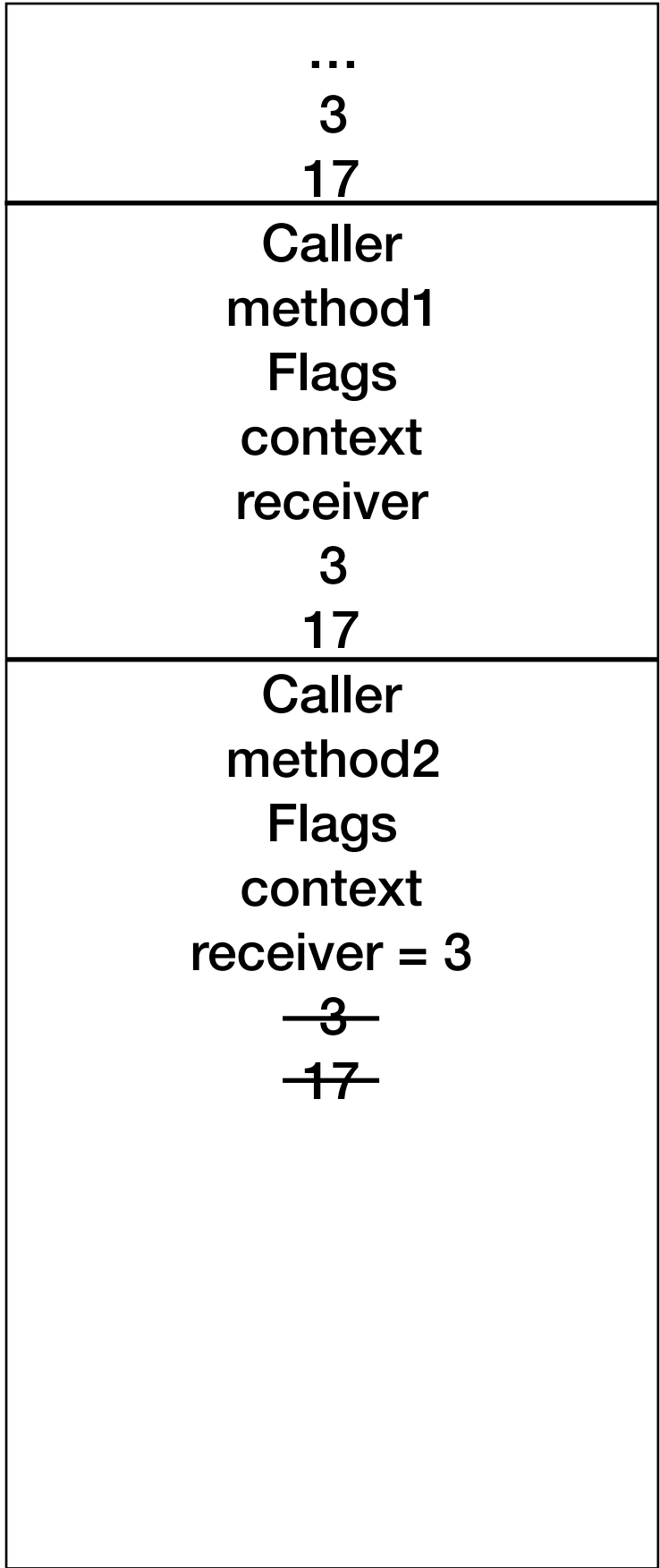
[stackIntegerValue:] Super and Self Messages sent but not implemented X ?



```
primitiveAdd
self pop2AndPushIntegerIfOK: (self stackIntegerValue: 1) + (self stackIntegerValue: 0)
```

1/3 [1] -- 1/3 [13]    arithmetic integer primitives    extension    F +L W

[pop2AndPushIntegerIfOK:] Super and Self Messages sent but not implemented X ?  
[stackIntegerValue:] Super and Self Messages sent but not implemented X ?



## Pop2....

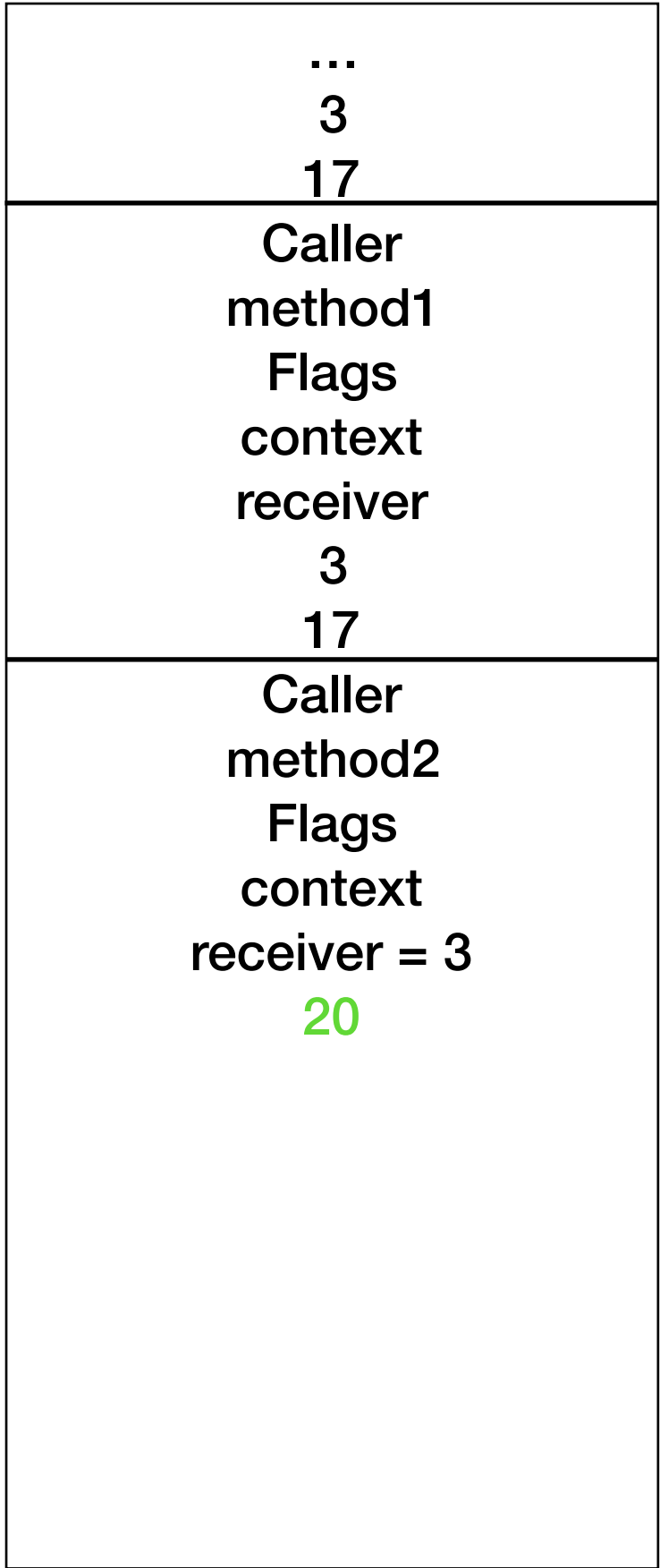
```

primitiveAdd
self pop2AndPushIntegerIfOK: (self stackIntegerValue: 1) + (self stackIntegerValue: 0)
  
```

1/3 [1] -- 1/3 [13]      × arithmetic integer primitives    extension    F +L W

❗ [pop2AndPushIntegerIfOK:] Super and Self Messages sent but not implemented ❗ ?

❗ [stackIntegerValue:] Super and Self Messages sent but not implemented ❗ ?



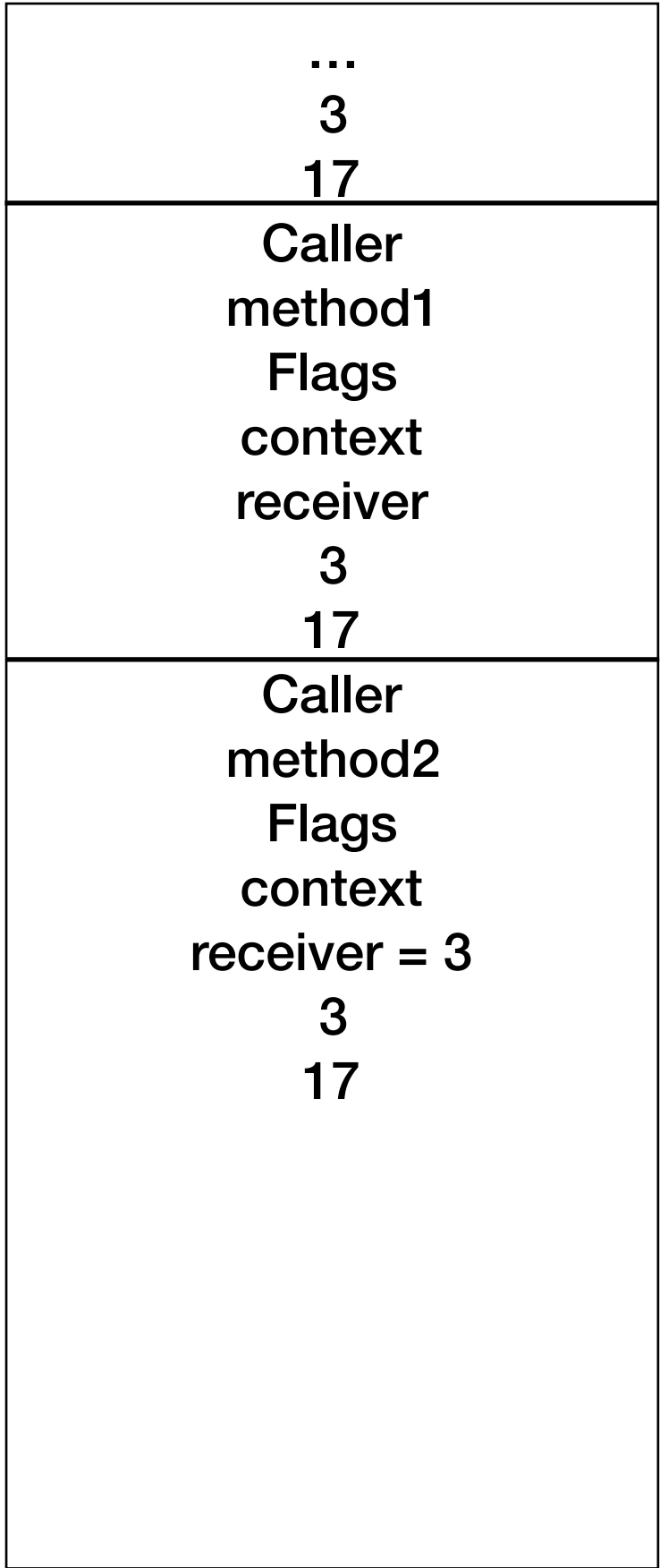
## ....andPushInteger

```

primitiveAdd
self pop2AndPushIntegerIfOK: (self stackIntegerValue: 1) + (self stackIntegerValue: 0)
  
```

1/3 [1] -- 1/3 [13]    arithmetic integer primitives    extension    F +L W

! [pop2AndPushIntegerIfOK:] Super and Self Messages sent but not implemented X ?  
 ! [stackIntegerValue:] Super and Self Messages sent but not implemented X ?



...ifOk:

**2 cases:**

**1. Primitive fail**

**2. Primitive succeed**

```

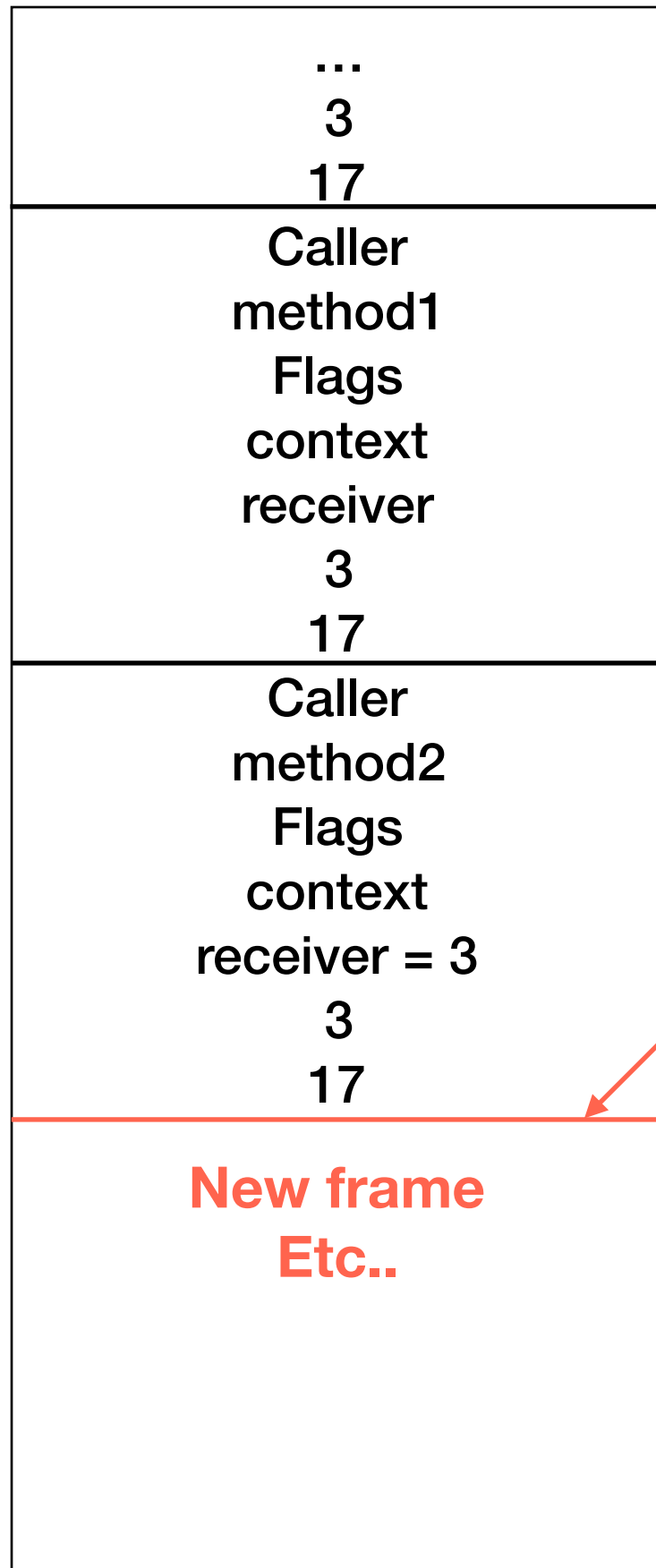
primitiveAdd
self pop2AndPushIntegerIfOk: (self stackIntegerValue: 1) + (self stackIntegerValue: 0)
  
```

1/3 [1] -- 1/3 [13]      arithmetic integer primitives   extension   F +L W

! [pop2AndPushIntegerIfOk:] Super and Self Messages sent but not implemented X ?  
 ! [stackIntegerValue:] Super and Self Messages sent but not implemented X ?



# Primitive fail



```
+ aNumber  
"Primitive. Add the receiver to the argument and answer with the result  
if it is a SmallInteger. Fail if the argument or the result is not a  
SmallInteger Essential No Lookup. See Object documentation whatIsAPrimitive."  
  
<primitive: 1>  
^ super + aNumber
```

Inspector on a CompiledMethod (SmallInteger>>#+)

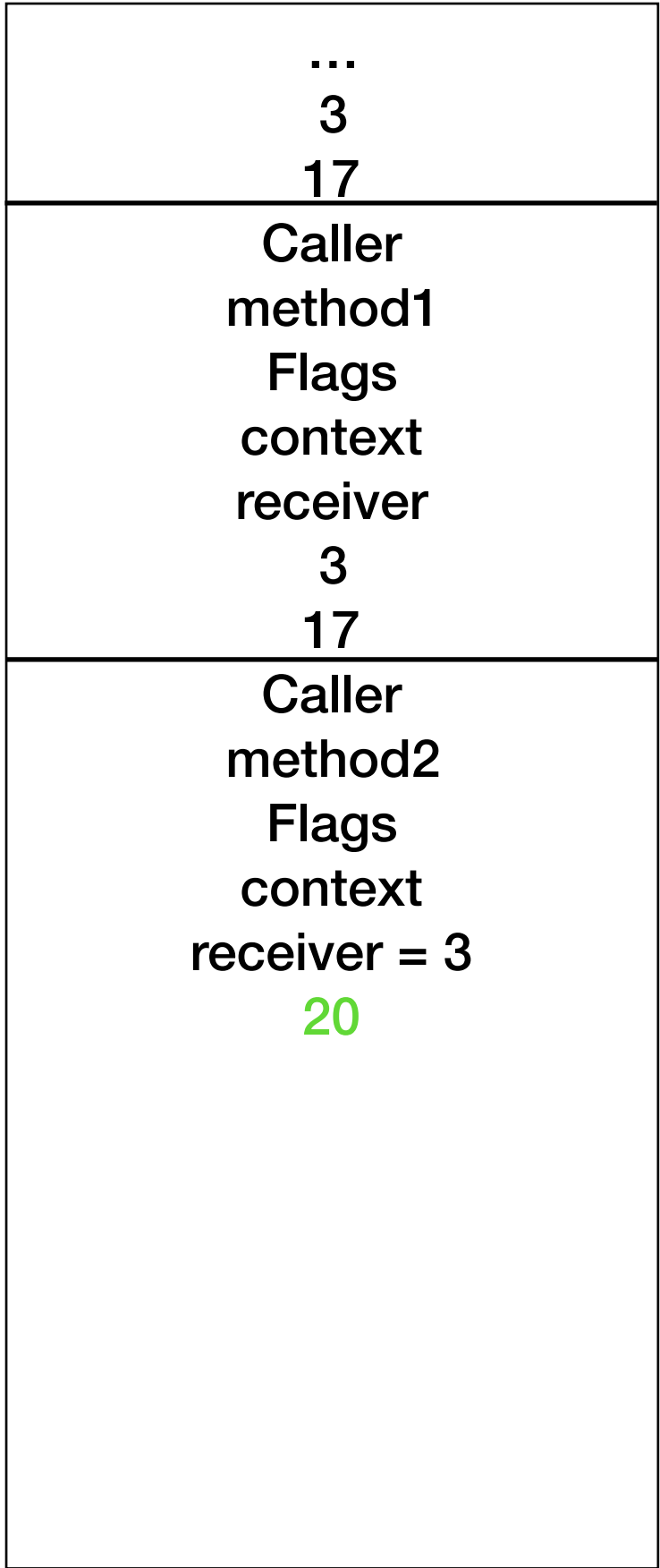
a CompiledMethod (SmallInteger>>#+)

Raw Source Pragmas **Bytecode** Ir AST Header Meta

```
33 <8B 01 00> callPrimitive: 1  
36 <70> self  
37 <10> pushTemp: 0  
38 <85 20> superSend: +  
40 <7C> returnTop
```

× arithmetic

**Execute fall back code**



```

+ aNumber
  "Primitive. Add the receiver to the argument and answer with the result
  if it is a SmallInteger. Fail if the argument or the result is not a
  SmallInteger Essential No Lookup. See Object documentation whatIsAPrimitive."

<primitive: 1>
^ super + aNumber

```

Inspector on a CompiledMethod (SmallInteger>>#+)

a CompiledMethod (SmallInteger>>#+)

Raw Source Pragmas Bytecode Ir AST Header Meta

```

33 <8B 01 00> callPrimitive: 1
36 <70> self
37 <10> pushTemp: 0
38 <85 20> superSend: +
40 <7C> returnTop

```

# Primitive succeed

primitiveAdd

```

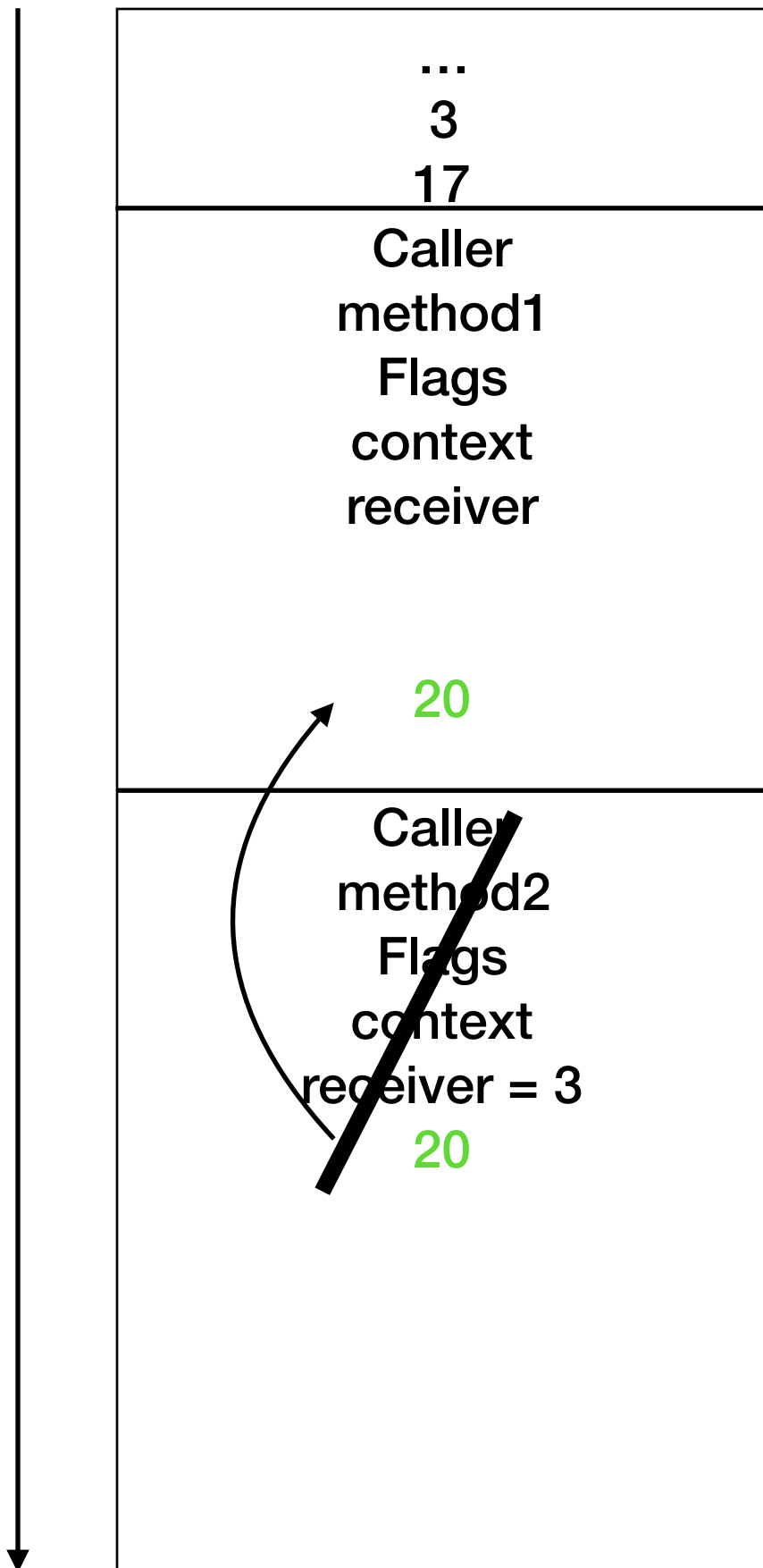
self pop2AndPushIntegerIfOK: (self stackIntegerValue: 1) + (self stackIntegerValue: 0)

```

1/3 [1] -- 1/3 [13]

[pop2AndPushIntegerIfOK:] Super and Self Messages sent but not implemented X ?

[stackIntegerValue:] Super and Self Messages sent but not implemented X ?



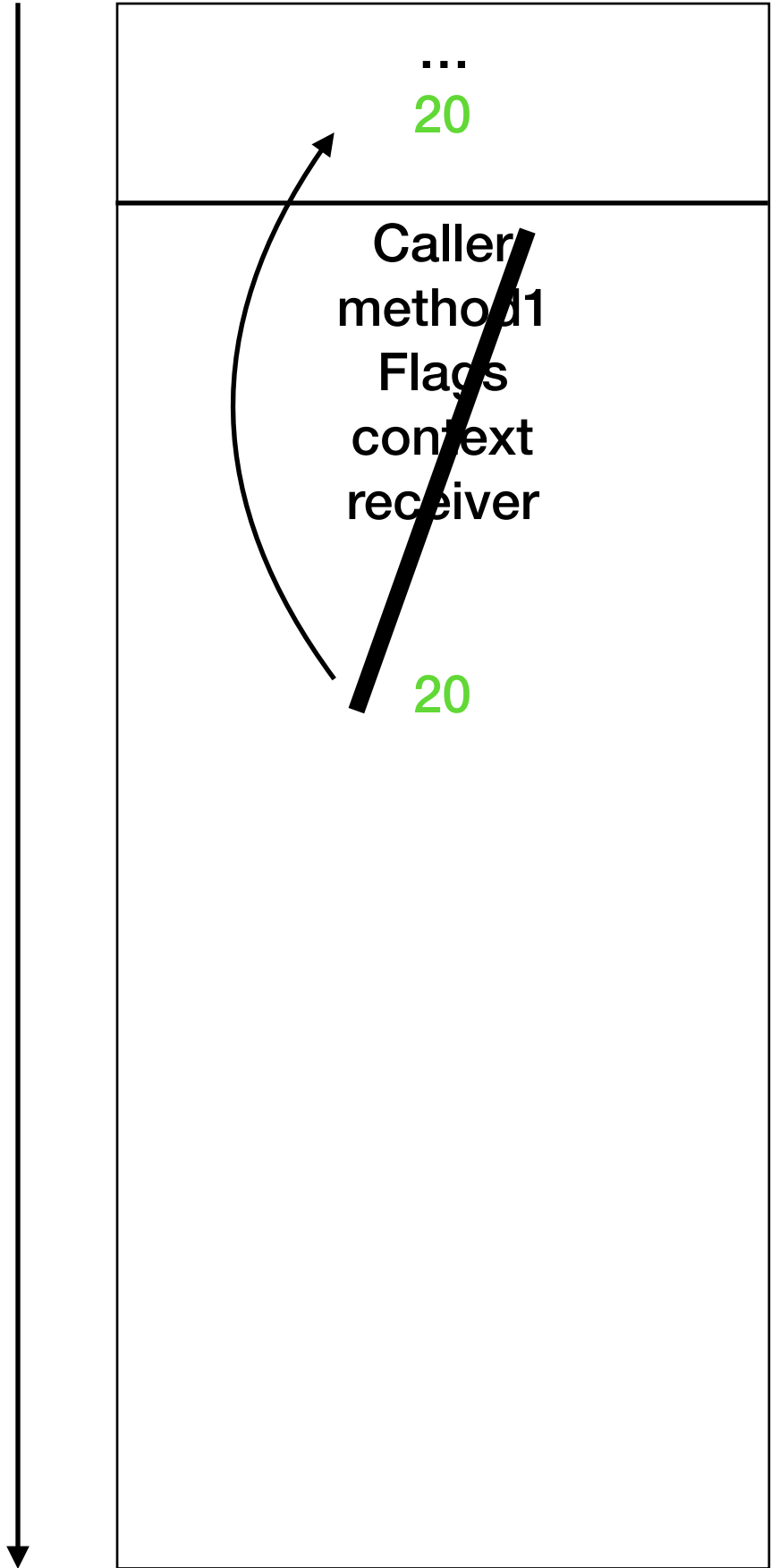
```

plus:
  plus: aNumber
    ^self + aNumber
  
```

Inspector on a CompiledMethod (Number>>#plus:)

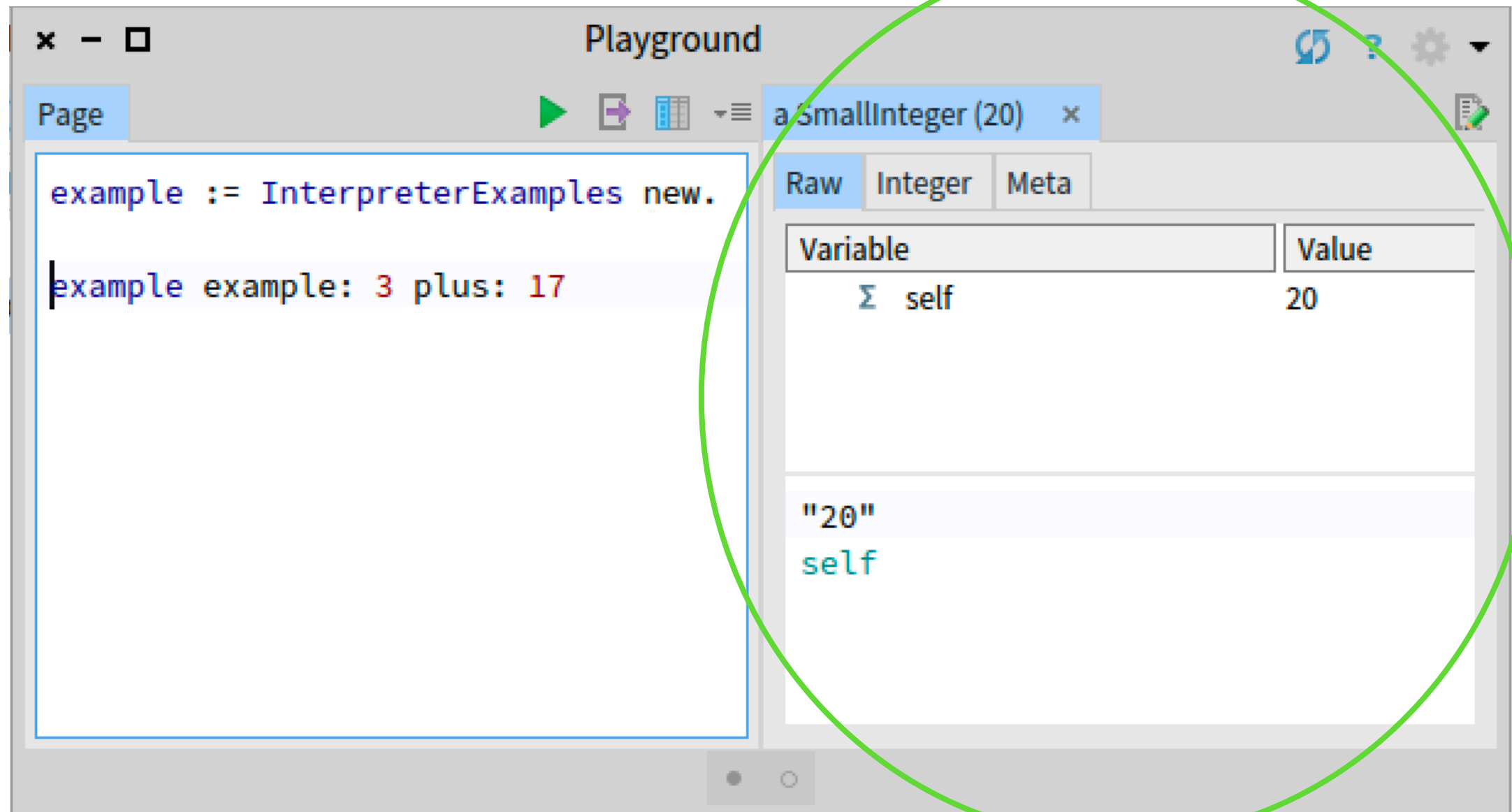
a CompiledMethod (Number>>#plus:)

Raw	Source	Bytecode	Ir	AST	Header	Meta
25	<70>	self				
26	<10>	pushTemp: 0				
27	<B0>	send: +				
28	<7>	returnTop				



```
Inspector on a CompiledMethod (InterpreterExamples>>#e
a CompiledMethod (InterpreterExamples>>#example:plus:)
Raw Source Bytecode Ir AST Header Meta
33 <10> pushTemp: 0
34 <11> pushTemp: 1
35 <E0> send: plus:
36 <7C> returnTop
```

# And we get



The screenshot shows a Scala Playground window with the following code in the editor:

```
example := InterpreterExamples new.  
example example: 3 plus: 17
```

The right-hand side of the playground displays the result of the execution, which is a `SmallInteger (20)`. A green circle highlights the inspection window for this object. The window has tabs for `Raw`, `Integer`, and `Meta`. The `Integer` tab is selected, showing a table with the following data:

Variable	Value
$\Sigma$ self	20

Below the table, the string representation of the object is shown as `"20"` and `self`.

## A more complex bytecode

```
? Comment x Interpreter x bytecodePrimAdd x + Inst. side method x
bytecodePrimAdd
| rcvr arg result |
rcvr := self internalStackValue: 1.
arg := self internalStackValue: 0.
(self areIntegers: rcvr and: arg)
  ifTrue: [result := (self integerValueOf: rcvr) + (self integerValueOf: arg).
           (self isIntegerValue: result) ifTrue:
             [self internalPop: 2 thenPush: (self integerValueOf: result).
              ^ self fetchNextBytecode "success"]]
  ifFalse: [successFlag := true.
            self externalizeIPandSP.
            self primitiveFloatAdd: rcvr toArg: arg.
            self internalizeIPandSP.
            successFlag ifTrue: [^ self fetchNextBytecode "success"]].
```

1/229 [1]

# Or function called by bytecodes

```
internalActivateN x
internalActivateNewMethod
| methodHeader newContext tempCount argCount2 needsLarge where |
<inline: true>

methodHeader := self methodHeaderOf: newMethod.
needsLarge := methodHeader bitAnd: LargeContextBit.
(needsLarge = 0 and: [freeContexts ~= NilContext])
  ifTrue: [newContext := freeContexts.
    freeContexts := self fetchPointer: 0 ofObject: newContext]
  ifFalse: ["Slower call for large contexts or empty free list"
    self externalizeIPandSP.
    newContext := self allocateOrRecycleContext: needsLarge.
    self internalizeIPandSP].
tempCount := (methodHeader >> 19) bitAnd: 16r3F.

"Assume: newContext will be recorded as a root if necessary by the
call to newActiveContext: below, so we can use unchecked stores."
where := newContext + self baseHeaderSize.
self longAt: where + (SenderIndex << self shiftForWord) put: activeContext.
self longAt: where + (InstructionPointerIndex << self shiftForWord)
  put: (self integerObjectOf: (((LiteralStart + (self literalCountOfMethodHeader: methodHeader)) * self wordSize) + 1)).
self longAt: where + (StackPointerIndex << self shiftForWord) put: (self integerObjectOf: tempCount).
self longAt: where + (MethodIndex << self shiftForWord) put: newMethod.
self longAt: where + (ClosureIndex << self shiftForWord) put: nilObj.

"Copy the receiver and arguments..."
argCount2 := argumentCount.
@ to: argCount2 do:
```

# Some references

**Smalltalk-80 bluebook, chapter on virtual machine implementation:  
<http://stephane.ducasse.free.fr/FreeBooks/BlueBook/Bluebook.pdf>**

**Eliott Miranda blogpost:  
<http://www.mirandabanda.org/cogblog/>**

**Test on VMMaker:  
<https://github.com/pharo-project/opensmalltalk-vm/tree/gc-tests>**

**VMLookUpTest  
VMPrimitiveTest  
VMByteCodesTest**